DISEÑO E IMPLEMENTACION DE UN SERVIDOR CON VARIOS SERVICIOS TCP/IP BAJO PLATAFORMA LINUX

JOSÉ ROBERTO BARRIOS ROCHA

INSTITUCIÓN UNIVERSITARIA TECNOLÓGICA DE BOLÍVAR

FACULTAD DE INGENIERIA DE SISTEMAS

CARTAGENA D. T. y C.

2001.

DISEÑO E IMPLEMENTACION DE UN SERVIDOR CON VARIOS SERVICIOS

TCP/IP BAJO PLATAFORMA LINUX

JOSE ROBERTO BARRIOS ROCHA
Código: 9605902

Trabajo de Grado presentado como requisito para optar al título de Ingenieros de Sistemas

Director:

GEOVANNI VASQUEZ
Ing. De Sistemas

Asesor:

ISAAC ZUÑIGA
Ing. De Sistemas

CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLÍVAR

FACULTAD DE INGENIERÍA DE SISTEMAS

CARTAGENA D. T. y C.

2001.

Nota de aceptación

_____

_____

_____

_____
Presidente del jurado

_____
Jurado

_____
Jurado

Ciudad y fecha (día, mes y año)

La Corporación Universitaria
Tecnológica de Bolívar, se reserva
el derecho de propiedad intelectual
de todos los trabajos de grado
aprobados y no pueden ser explotados
comercialmente sin su autorización.

A todas aquellas personas que no creyeron en mí, a los que me criticaron y por ende me hicieron una mejor persona. A mi padre que con hechos me enseño a luchar con un objetivo, a mi abuela que me aconsejó en todo momento, a mi tía que en cualquier circunstancia a estado a mi lado, a mi hija Valentina que me alegró en mis momentos de tristeza, a mis amigos y a Laurie que estuvo a mi lado desde el comienzo.

AGRADECIMIENTOS

El autor expresa sus agradecimientos a:

Ing. Geovanny Vásquez. Ingeniero de Sistemas y Director de la investigación, por sus valiosas orientaciones y consejos.

Ing. Isaac Zúñiga Silgado. Ingeniero de Sistemas y Asesor de la investigación, por su constancia y motivación para el desarrollo de este trabajo.

Ing. Gonzalo Garzón. Ingeniero de Sistemas, por su valiosa colaboración y constante empeño para la realización de la investigación.

CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

LISTA DE ANEXOS

GLOSARIO

ACK: es el acuse de recibo en el canal de comunicación, es el que determina si se recibió el paquete de información ó no.

ALLOC: es la función para la asignación de parte de memoria.

APACHE: es el nombre de nuestro servidor web.

Browser: explorador ó navegador, es el programa cliente que permite ver páginas web con los aditivos que ella lleva.

Cabecera: es el primero número de bytes de un archivo, paquete, datagrama u otro objeto utilizado, generalmente lleva un formato especial ó son los primeros bytes físicos de el objeto.

CERN: es un servidor web.

CGI: (Common Gateway Interface), es un lenguaje de programación que se usa para dar resultados al cliente procesados por el servidor.

CHGRP: “chgrp” comando en UNIX para cambiar el grupo a un archivo.

CHMOD: “chmod” comando UNIX para cambiar el modo de un archivo.

CHOWN: “chown” comando UNIX para cambiar de propietario a un archivo.

DB: (DataBase), base de datos.

DNS: (Domain Name Server), servidor de nombres de dominio, es el que se encarga de la administración de los nombres en una red así como de sus direcciones IP para proveer el acceso.

Demonio: proceso que siempre se estará ejecutando en un sistema operativo.

FILESYSTEM: sistema de archivos, es la estructura básica de todo sistema operativo, donde se maneja las estructuras de los archivos.

FORK: “fork”, comando UNIX que sirve para la clonación de procesos.

FTP: (File Transfer Protocol), protocolo de transferencia de archivos.

GID: (Group IDentifier), propiedad de los archivos para la identificación del grupo.

GIF: (Graphic Interface Format), formato de archivo de imagen / animación.

GNOME: gestor de ventanas para la consola gráfica de LINUX.

GRUB: (GRand Unified Bootloader), es el iniciador para algún sistema operativo.

HTML: (Hiper Text Manager Language), es el formato hipertexto con el que se realizan las páginas web.

HTTP: (Hiper Text Transfer Protocol), es el protocolo de transferencia de hipertexto, hoy en día usado para un gran número de usos.

HTTPD: “httpd”, demonio del protocolo http.

IP: (Internet Protocol), protocolo de Internet. Dirección IP es el dirección física de una máquina de esta manera se diferencia una máquina de otra en una red.

INETD: servicio LINUX, que permite ofrecer diferentes servicios TCP/IP.

KDE: gestor de ventana para nuestra consola gráfica de LINUX.

LILO: (LInux LOader), es un iniciador de sistema operativo.

LINUXCONF: “linuxconf”, utilidad para la configuración de nuestra máquina LINUX.

LISA: Utilidad que viene con la mayoría de las distribuciones de LINUX, que permite facilitar la instalación y configuración del sistema.

LOG: archivo donde se almacena errores, seguimiento y desempeño de una máquina, un servicio ó proceso.

MAIL: servicio de correspondencia entre usuarios.

MALLOC: “malloc”, comando para la asignación de memoria.

MD5: (Message Digest), algoritmo de cifrado por autenticación de usuario.

MM: Biblioteca de funciones para la utilización de módulos en nuestro servidor Apache.

NETSCAPE: marca fabricante del navegador Netscape, así como gran variedad de utilidades cliente/servidor para manejo de la internet.

PERL: lenguaje de programación, para realización de utilidades e inclusiones en el lado del servidor.

PHP: (HiPertext Preprocessor), lenguaje de programación embebido en HTML para la realización de Server Sides.

PID: número identificador de un proceso.

POSTGRES: base de datos, gratuita que generalmente viene con las distribuciones de LINUX.

PROFTP: servidor FTP.

PROTOCOLO: conjunto de reglas y condiciones que determinan un sistema de comunicación y servicios en la red. Protocolo propiamente en su sentido de la palabra es un conjunto de reglas que se disponen para un asunto ó hecho.

PROXY: Programa usado para aumentar la seguridad de una red, para administrar el ancho de banda ó para servir documentos de un caché.

QMAIL: servidor para el servicio de correspondencia.

QUOTA: límite que se le asigna a los usuarios para el tamaño de información guardada.

REYSERFS: tipo de sistema de archivo para sistemas UNIX.

ROOT: “root”, usuario principal de todo sistema UNIX. Directorio root ó raíz (/), es el directorio principal del sistema de allí se desprenden ramas que representan

directorios y/ó particiones en nuestro sistema.

RSA: sistema de encriptación de clave pública.

SENDMAIL: servidor de correspondencia.

SERVER SIDE: inclusión en el lado del servidor, son programas que debe ejecutar el servidor para enviarle el resultado al cliente.

SGID: comando que se incorpora para la ejecución de programa, para que se ejecute siempre con permisos del grupo del súper usuario.

SSL: (Secure Socket Layer), sistema de comunicación por socket seguro. Módulo del Apache para entablar comunicación segura entre seguros.

STDERR: archivo de salida de errores para un proceso en LINUX.

STDIN: archivo de entrada para un proceso en LINUX.

STDOUT: archivo de salida para un proceso en LINUX.

SUID: comando que se incorpora para la ejecución de programas, que se ejecuten siempre con permisos del root.

TAG: comando pequeño para el formato hipertexto, este permite agregar propiedades y asignaciones a un hipertexto.

TCP: (Transfer Control Protocol), protocolo para el control de transmisiones.

TCP_WRAPPER: es una herramienta para el control de servicios.

TOKEN: comando ó palabra.

UID: (User IDentifier), identificador de usuario de un archivo ó proceso.

URL: (Uniform Resource Locators), localizador de informe de recursos, dirección de archivos en la red.

Web Server: servidor web, servidor de transferencia de hipertexto y archivos.

INTRODUCCIÓN

Con el inicio de la Internet se ha desarrollado gran variedad de servicios sobre la familia de protocolos TCP/IP, es deber de todo profesional de la informática conocer detalladamente los servicios más usados, tanto como cliente como servidor.

Al seguir las páginas de este documento encontrará que no parece un documento de un proyecto sino un manual, en si es lo que esperaba entregar un manual que enseñe, un manual que explique y que trate de mostrar distintos aspectos de los servicios ofrecidos por la familia de protocolos de TCP/IP.

Como todo proyecto el comienzo es la instalación y descripción del sistema operativo usado, se trató de buscar la manera de estándarizar la instalación para que pudiera servir para cualquier distribución de Linux tipo Red Hat, como hay que notar en la red existen diversos tipos de Linux, los más usados son Linux Red Hat y Linux Debian. La incorporación de los servicios TCP/IP en este sistema, hacen que el usuario regrese para asociar comandos con referencias a los items tratados para asegurar la seguridad y eficiencia. Luego se trata de colocar el diseño,

instalación y configuración de los servicios más importantes, de modo que los temas fueran útiles para el desarrollo.

Los capitulos están distribuidos de la siguiente forma, el primero nos habla de lo concerniente al sistema operativo, el segundo habla sobre la red TCP/IP y la incorporación de esta al sistema ya tratado, los capitulos siguientes son ordenados por servicios, donde detallamos cada servicio, hacemos una breve descripción del diseño que este debe tener y la implementación bajo la plataforma seleccionada.

Notarán una especie de desorden en algunos capitulos, hecho intencionalmente para comprender más fácilmente los mecanismos de instalación y configuración de cada servicio, (no todos se instalan ó configuran igualmente).

Para cada servicio se trata de profundizar en los conceptos de seguridad y desempeño, sin centrarse mucho en el tema, para no salirse de los objetivos fijados.

Podrá usted darse cuenta al recorrer las páginas de este documento que pueden salir varios proyectos de tesis, de cada uno de los servicios, porque el ser todos

modulares podemos configurar, crear, cambiar, en fin gran cantidad de opciones para crear los servicios exactamente como lo necesitamos.

La gran mayoría de scripts y comandos de ejemplos son ejecutados desde la máquina lhost con usuario root; los archivos de configuración para algunos servicios generalmente están hechos para el equipo sistemas.cutb.edu.co. Los scripts y comandos son especificados en letra distinta lo mismo que los archivos.

Hay que entender algo, todos los archivos de configuración de Linux no son iguales, mientras que en uno los comentarios son (#), en otros son antecedidos por (;). Al configurar un servicio lea detalladamente las instrucciones, ya que una inadecuada configuración puede inutilizar completamente el servicio y/ó la máquina.

Este documento no presenta la panacea a los males de instalación de servicios ni es tampoco la versión actual de todos los servicios aunque así se haya querido, es tarea de usted administrador, ingeniero, estudiante, maestro, buscar en la red aquella versión actualizada que pueda mejorar la anterior ya que la actualización de mucho software, parches y nuevos productos son tema diario de noticia, así

pues lo invito a probar, investigar, y crear. Solo de esta manera podrá acceder a gran parte de los servicios que podemos prestar desde un servidor.

El proyecto se límito a configuraciones básicas de los servicios, el tema da para hacer mucho por ende, este documento puede ser complementado con otros documentos y proyectos que se han realizados para mayor comprensión.

Por algunas razones que se ha dicho debe darse cuenta que el protocolo TCP/IP ofrece gran cantidad de especializaciones desde su nivel físico hasta su nivel de servicios por ende tratar de acaparar todads estas especialidades en un solo documento es algo prácticamente imposible, pero un pie al cimiento puede llevar a construir un gran edificio.

Podemos decir que este proyecto abarca temas importantes ya que las computadoras hoy en día no se situan solas sino por el contrario se convierte en una gran red que llamamos la Internet, que es prácticamente la integración de computadoras independientes a su arquitectura, sistema operativo ó tipo de conexión, todo esto gracias a la gran familia TCP/IP.

Se espera que el documento llene las expectativas acordadas.

1. USO DE UN SISTEMA OPERATIVO MULTIUSUARIO

Para poder ofrecer un servicio se hace indispensable tener un sistema operativo multiusuario, el cual pueda ofrecer un servicio a cada usuario independientemente.

La diferencia entre un sistema de archivos multiusuario y un monousuario radica en que un sistema operativo multiusuario tiene la oportunidad de que varios usuarios interactúen con él al mismo tiempo, así todos pueden acceder a sus servicios independientemente y el sistema operativo a la vez puede manejar el nivel de seguridad entre ellos, de esta manera cada usuario tiene permisos asignados sobre archivos y procesos lo mismo que puede mantener la seguridad de sus documentos. Así un sistema operativo multiusuario puede compartir ó puede negar.

Sí nosotros deseamos implementar servicios de red en una máquina debemos primero tener un sistema operativo multiusuario, en la actualidad existen muchos sistemas operativos multiusuarios como WINDOWS NT, WINDOWS 2000, Novell Netware, UNIX, LINUX, FreeBSD, entre otros, donde cada uno de ellos puede implementarse con sus servicios de red.

1.1 DISEÑO PARA UN SISTEMA OPERATIVO MULTIUSUARIO

Cada sistema operativo posee un distinto método para trabajo, vamos a referirnos al modo y diseño del sistema operativo LINUX, que vale también para otros sistemas tipo UNIX.

Antes que nada debemos saber como trabaja el sistema operativo.

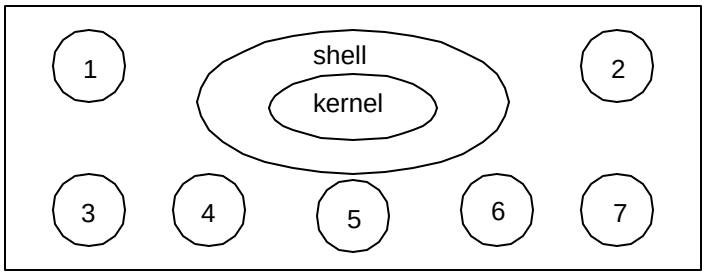

Figura1. Arquitectura de LINUX.

Donde 1,2,3,4,5,6,7 son los tipos de utilidades en nuestro sistema operativo, el kernel es el núcleo de nuestro sistema, es el encargado de manejar el hardware, decide que procesos se ejecutan, también crea y mantiene archivos del sistema con los permisos asociados a ellos. Todos los programas que trabajan con LINUX se comunican con el kernel a través de comandos de bajo nivel llamados *SYSTEM*

CALLS. Así mismo el kernel es el encargado de manejar los dispositivos de entrada y salida como también los mecanismos de protección.

El shell se encarga de administrar los recursos y usuarios. Es un interpretador de comandos que interactúa con el usuario. El shell se activa cuando comienza una sesión de trabajo (LOGIN) y se desactiva cuando se termina la sesión de trabajo (LOGOUT). En LINUX podemos observar distintos tipos de shell entre ellos *sh, bash, tcsh, csh, pdksh, zsh, ash* y *mc*.

Las utilidades son módulos adicionales desarrollados para darle mayor capacidad y eficiencia al sistema, entre ellos calculador de operaciones aritméticas, compilador y depurador de C, herramientas para el desarrollo de software, controladores de dispositivos, editores de texto, comunicaciones y manejo de redes, manipulador de archivos y cadenas, utilidades para terminales e impresoras, etc.

LINUX es un sistema operativo tipo UNIX, por ende entiende todo como si fuera archivos, un directorio es un archivo que contiene la información de otros archivos, un proceso es un archivo, un dispositivo es un archivo y así sucesivamente. Por

esto mismo LINUX trabaja en disco y se hace necesario crear una partición swap que servirá como partición de intercambio de memoria para los procesos.

Antes de entrar a la parte de administración de LINUX es importante aclarar la organización de los directorios en LINUX. Por lo que ya hemos dicho que LINUX es un sitema tipo UNIX, podemos observar que su organización de directorios de LINUX es completamente igual a la de UNIX, en la gráfica siguiente podemos observar la estructura de directorios de LINUX.

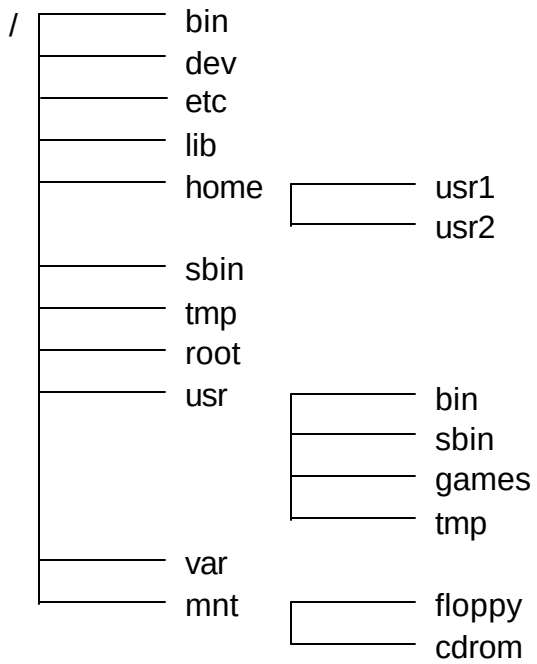

Figura 2. Estructura de directorios

Linux parte de la estructura / que es el directorio raíz es el inicio de nuestro árbol de directorios. Ahora bien, como sabemos Linux no posee unidades como Windows, sino que se representan por directorios, la separación entre directorios y ficheros es por "/" no "\", además por ser multiusuario de acuerdo al usuario y su privilegio puede acceder a ciertos directorios.

El mayor de todos es el usuario root ó raíz que es el único que puede acceder al directorio raíz de Linux; luego siguen los superusuarios que poseen un directorio generalmente con su nombre, en el directorio home, y por último los usuarios del sistema que tambien poseen un directorio que se encuentra en home y allí conservar sus datos, por esta medida podemos controlar los programas que puedan efectuar los usuarios para que estos no afecten a nuestro sistema.
A continuación una pequeña explicación de cada directorio y que contiene:

- */bin*: podríamos decir que son los binarios de uso básico de nuestro sistema GNU/Linux. Aquí podemos encontrar varios comandos, como por ejemplo el comando mount.

- */dev*: antes hemos dicho que en el sistema UNIX, "todos son archivos", quiere decir que en nuestro sistema GNU/Linux todo queda representado como fichero, así si queremos imprimir algo lo hacemos en el fichero */dev/lp1* (si nuestra impresora la hemos configuramos allí), el kernel se encargará de enviarlo a la impresora que se encuentra en ese dispositivo, así mismo el disco duro primario se representa por */dev/hda* y sus particiones */dev/hda1, /dev/hda2*, etc.. La unidad de CD que generalmente está como esclavo se encuentra en */dev/hdb*. Así en este directorio se encuentra los discos, la unidades zip, puertos, tarjetas de red,etc.. Recordemos que todos se emplean como ficheros y los directorios no son más que ficheros que contienen la información del resto de los archivos, que tienen bajo su nivel. También podemos mencionar el dispositivo */dev/null*, que nos redirecciona la salida de un comando a un "agujero negro" donde mandar los procesos que queramos que tengan salida a este dispositivo.

- */sbin*: este directorio contiene los archivos básicos binarios de nuestro sistema GNU/Linux.

- */usr*: aquí es donde suelen instalar las aplicaciones que no están en el sistema más básico, como pueden ser los entornos gráficos, las aplicaciones, los juegos, etc.. Este directorio tiene también una estructura jerárquica, con los binarios en

/usr/bin, en los juegos */usr/games*, las librerias que usan */usr/lib*. Aquí también se encuentran, en un principio, las páginas del manual (man).

- */lib*: en este directorio podemos encontrar las librerias básicas que necesita un sistema GNU/Linux para funcionar.

- */mnt*: como hemos visto en el directorio */dev* los discos son tratados como archivos y no como dispositivos físicos, pero para acceder a ellos de esta forma hay que "montarlos" antes de acceder a ellos. Este procedimiento es ejecutado mediante el comando mount, para poder decirle al sistema que en ese dispositivo podemos trabajar. Y para llevar un orden generalmente tenemos un directorio que contiene un estructura de floppy y cdrom, pero el montaje puede ser en cualquier directorio por esa razón debemos mantener la seguridad accediendo privilegios al montaje de unidades y el directorio a usar para ello.

- */tmp*: este directorio contiene información temporal que es eliminada al nuevo arranque del sistema.

- */etc*: este directorio es usado por el sistema para guardar sus ficheros de configuración. El nombre ocurrio cuando UNIX en su afán de organizarlo todo los

ficheros de configuración quedaron en un directorio llamado */etc*, así que lo dejaron así.

- */home*: este es el directorio de trabajo de los usuarios generalmente debe estar en una partición aparte ó en una unidad propia, como también el directorio */usr*. Este directorio también puede conservar los archivos de nuestros servicios HTTP y FTP. Es recomendable hacer varias particiones con varios */home, /home1, /home2*, etc.. para separar la información de cada directorio para evitar que otros usuarios puedan observar información que no estén privilegiados.

- */var*: contiene archivos que cambia cuando el sistema trabaja normalmente, es usado por la administración y las cuentas.

- */root*: es el directorio de trabajo del administrador del sistema, que esta separado del directorio /home por obvias razones de seguridad.

1.2 ADMINISTRACION DEL SISTEMA LINUX

Cada sistema UNIX se administra en forma distinta a otros sistema, LINUX no es la excepción a esa regla, aún así el papel de la administración difiere dependiendo

del sistema que se tiene. La importancia de la administración del sistema determina la seguridad y desempeño que pueda tener el sistema, el coste de una mala administración recae sobre el administrador del sistema; de acuerdo al tamaño del sistema así será el número de administradores de sistema y cada uno administrará una parte de este. En el mundo actual donde las computadoras son un entorno grupal y no un ente individual se hace necesario tener dos administradores, el administrador del sistema y el administrador de red.

En caso de tener instalada una base de datos se hace necesario poseer un administrador de base de datos.

1.2.1 Administrador del sistema: El administrador del sistema es la persona encargada de proporcionar a los usuarios un entorno seguro, eficiente y fiable. Tiene el poder y la responsabilidad de establecer y mantener un sistema que facilite un servicio fiable y efectivo.

Todos los sistemas LINUX tienen un único usuario que posee los permisos para realizar cualquier operación en el sistema, este super-usuario tiene una entrada especial denominada “root” y tiene su directorio de origen */root*, la contraseña para este usuario se ingresa desde la instalación del sistema y debe ser cambiada cada

cierto período dependiendo de la seguridad que se desee en el sistema. El nombre de entrada del superusuario (root) se utiliza sólo en algunos casos y debe ser el menor número posible (dos ó tres como mucho).

Cuando un usuario ingresa como *root*, tiene el control de manipular el sistema como le plazca, por esta razón el administrador tiene que conocer muchos aspectos técnicos del sistema.

También tiene que conocer las necesidades de los usuarios, así como el objetivo del sistema. Cualquier sistema informático es un recurso limitado, por lo que deben establecerse y cumplirse un cierto número de normas que regulen su uso.

Las responsabilidades del administrador del sistema generalmente son estipuladas por los módulos administrativos de la empresa, sin embargo existen una serie de tareas que los administradores deben gestionar como son:

- Administrar usuarios: Añadir usuarios, eliminarlos y modificar sus posibilidades y privilegios.

- Configurar dispositivos. Hacer disponibles y compartir dispositivos, como por ejemplo impresoras, terminales, modems, unidades de cinta, etc.

- Hacer copias de seguridad. Programar, hacer y almacenar copias de seguridad para restaurarlas en caso de que se dañen o pierdan archivos del sistema.

- Apagar el sistema. Apagar el sistema de un modo ordenado para evitar inconsistencias en el sistema de archivo.

- Formar usuarios. Proporcionar directa ó indirectamente la información necesaria para que los usuarios puedan utilizar el sistema de forma efectiva y eficiente.

- Asegurar el sistema. Evitar que los usuarios interfieran unos con otros accidental ó deliberadamente.

- Registrar los cambios del sistema. Mantener un libro para registrar cualquier actividad significativa que se refiera al sistema.

- Asesorar a los usuarios. Actuar como un "experto" para ayudar a los usuarios.

Es caso que el administrador sepa las distintas formas de modelamiento de procesos, existe el modelo de procesos centralizado donde las operaciones son

realizadas por un servidor y los usuarios acceden por medio de terminales brutas ó terminales "inteligentes", las primeras son terminales que solo poseen pantalla y teclado, cualquier proceso es llevado a cabo desde el servidor; las segundas se refieren a terminales que pueden tener mayor cantidad de hardware por lo tanto pueden ejecutar pequeñas operaciones; la manera de comunicación entre estas máquinas y el servidor es a través de los puertos.

El proceso distribuido es mas complejo debido a que la realización de procesos es llevada a cabo desde terminales de trabajo, dejando al servidor el servicio de archivos y la ejecución de tareas administrativas; el entorno físico de una red distribuida es mas compleja y posee distintos tipos de topología y medio físico para implementarlo.

Cada sistema de red distribuida tiene su ventaja y desventaja uno sobre otro. Dada la importancia de este ítem es deber del administrador estar desde el comienzo del diseño para evaluar cada una de las partes del diseño e implementación.

1.2.2 Administrador de la red. Una red LINUX normalmente tiene la forma de muchas computadoras grandes y pequeñas, unidas a través de cables y

conectadas directa ó indirectamente por líneas a la INTERNET. La administración de la red es llevada a cabo por uno ó más administradores ubicados en el centro de la red.

Al tener varios sistemas UNIX/LINUX conectados en red, es aconsejable tener un administrador de la red. El prospecto debe contar con conocimientos de conexión de sistemas (LAN ó modems), que sepa asignar el nivel de seguridad requerido y distribuir los periféricos compartidos. En el trabajo diario el administrador de la red es quien se ocupa de la lista de nombres del sistema, de las direcciones de red, del acceso a usuarios de forma general, quien se asegura de que la red esta funcionando adecuadamente.

Si una red posee varios servicios es recomendable que exista un administrador para cada uno de ellos.

Las tareas comunes de un administrador de la red podemos citarlas como:

- Configuración del sistema. El administrador de la red debe estar capacitado para configurar las máquinas que puedan ingresar a la red, debe probar cada uno

de los servicios que pueda usar así como la comunicación ya sea por cableado ó por conexión telefónica.

- Manejo de periféricos. El administrador de la red debe tener en perfecto estado el desempeño de sus dispositivos periféricos como impresoras y unidades de disco, este debe mantener el nivel de seguridad y los permisos para uso de cada uno de los dispositivos periféricos que estén en la red.

- Supervisión del sistema. Luego de la instalación del sistema el administrador puede usar algunas herramientas de LINUX para medir el desempeño del sistema. La supervisión en sistemas LINUX es un trabajo continuo, pero la carga de trabajo debe estabilizarse luego de un tiempo, más si no hay una continua instalación de software ó hardware, el buen administrador puede prever errores ó conflictos con la constancia de la supervisión y así poder determinar el tipo de error que se pueda tener.

- Hacer frente a las actualizaciones. La mayoría de software hoy en día puede actualizarse automáticamente, es deber del administrador elegir la versión correcta del nuevo paquete, en ocasiones es mejor utilizar la versión no critica y esperar los comentarios y bugs que pueda tener, en otras ocasiones es necesario

la instalación de estas nuevas actualizaciones para evitar algún daño severo (es el caso de los virus informáticos).

Dadas ya las tareas que debe cumplir un administrador de la red es necesario saber la formación que deba tener este:

- Diseño y utilización de LINUX. El administrador debe conocer el entorno LINUX, así como temas explícitos como redireccionamientos, conducciones, procesos de fondo, etc..

- El editor vi. En la mayoría de sistemas UNIX, existe un editor de texto llamado vi (VIM en las distribuciones de LINUX), es necesario que el administrador lo conozca debido a que es el denominador común.

- Programación con secuencias de SHELL. Muchos de los programas utilizados para administrar sistemas UNIX están escritos en lenguajes de secuencias de SHELL y puede ser necesario modificarlos para adaptarlos a las necesidades específicas de instalación. Hay que tener en cuenta la variedad de SHELL que existen así pues se hace necesario que sepa que tipo de SHELL usar. Es necesario también que conozca el lenguaje de administración de sistemas PERL,

puesto que este dispone un conjunto de herramientas muy sólidas para la administración de sistemas en un entorno de programación.

- Comunicaciones. El administrador debe tener bases sólidas sobre los conceptos de redes. Debe tener dominio de TCP/IP y los protocolos relacionados. Conocimiento del protocolo PPP para una conexión a INTERNET asincrónica. Seria ideal la disponibilidad de un laboratorio de pruebas para el uso de estos protocolos con el mayor número de opciones posibles para mejorar el sistema que se posee. A la vez debe tener una formación actualizada sobre los nuevos acontecimientos al respecto.

- Convenciones UNIX. El administrador debe conocer plenamente el entorno tipo UNIX, su organización y trabajo para mejorar la gestión y versatilidad de nuestro sistema.

1.2.3 Administración de cuentas. En un sistema LINUX existen distintas maneras de agregar usuarios es tarea del administrador agregar y administrar el grupo y usuario.

Es bien sabido que sólo root tiene el permiso para agregar usuarios y grupos a todo el sistema, el usuario sólo tendrá acceso al los archivos y permisos del grupo

y el solamente podrá asignar acceso y permiso a sus propios archivos. Por ende el usuario root debe ser usado con fines específicos al mantenimiento y configuración del sistema, ya que un mal uso de esta cuenta puede llevar a daños irreversibles a nuestro sistema.

Para una correcta administración de cuentas debemos tener en cuenta estos pasos:

- Política de cuentas. Una cuenta consta de dos elementos autorización para iniciar una sesión y autorización para acceder a los servicios. La autorización para iniciar una sesión es un privilegio que no hay que conceder a la ligera. Sí es posible proporcionar a los usuarios servicios críticos sin concederles acceso a la shell, hágalo, el conceder el permiso de shell a usuarios remotos puede aparecer inmediatamente una brecha de seguridad. En caso de que los usuarios deban ingresar por medio de una shell al sistema puede tomar estas medidas que reducirán riesgos:
    - Dedique una máquina exclusivamente para el acceso al shell.
    - Restrinja dicha máquina solamente para el uso de la shell.
    - Elimine de ella todos los servicios de red que no sean indispensables.

- Instale un conjunto genérico de aplicaciones y al crear particiones, tenga en cuenta el reabastecimiento tras algún desastre. Las máquinas con la shell no suelen recibir muy buen trato y de allí la reinstalación continua del sistema.
- Prohíba las relaciones de confianza entre la shell y otras máquinas.
- Considere la posibilidad de separar los sistemas de archivos importantes en otras particiones, mueva los archivos binarios SUID a una partición que LINUX monte el no setuid.
- Redirija los registros a un servidor de registros ó a algún medio donde solo pueda escribirse y registre todo.
- Calcule la cantidad de usuarios que puedan tener acceso a la shell y cuántos puedan ocasionarle un daño al sistema.

➢ Estructura de cuentas. Una cuenta en su sentido más específico cuenta con un nombre de usuario y una contraseña válidos, un directorio inicial y un acceso a la shell. Cuando un usuario intenta iniciar una sesión en LINUX verifica si se cumplen los requisitos buscando en el archivo *passwd*. El archivo passwd se encuentra en el directorio */etc*, este archivo posee la información de cuentas por medio de registros donde cada registro posee el formato:

`nombre_usuario:contraseña_cifrada:ID_usuario:ID_grupo:Nombre_real:directorio_principal_usuario:shell_usuario`, un ejemplo de cómo se vería el usuario root en el archivo *passwd* sería: `root:x:0:0:root:/root:/bin/bash.`

- Creacion y eliminación de cuentas: Como sabemos existen muchas formas de crear cuentas. Existen muchas utilidades gráficas y utilidades en cada sistema LINUX que se pueden usar para crear ó eliminar cuentas, como es el caso del linuxconf, lo lamentable de este caso es que proporcionamos un hueco para que intrusos puedan afectar nuestro sistema.

Para la creación de cuentas manualmente seguimos los pasos siguientes:

- Editamos el archivo */etc/passwd* y dado el formato anterior agregamos una línea al archivo donde detallaremos la información de los usuarios.
- Creamos el directorio donde residirá nuestro usuario, este directorio deberá ser creado en la ruta que se especifico en el archivo *passwd*. De la manera siguiente: `mkdir /home/nuevo_usuario.`
- Copiamos los archivos de */etc/skel* al directorio del nuevo usuario por el comando desde shell, `cp /etc/skel/.* /home/nuevo_usuario/`
- Modificamos los permisos hacia este directorio así:

```
chown nuevo_usuario /home/nuevo_usuario
chown nuevo_usuario /home/nuevo_usuario/.*
chgrp nuevo_usuario  -id_usuario /home/nuevo_usuario
chgrp /home/nuevo_usuario
chmod 755 /home/nuevo_usuario
chmod 644 /home/nuevo_usuario/.*
```

Para la eliminación de usuarios basta con eliminar sus líneas del archivo */etc/passwd* y remover el directorio con el comando:
`rm -r /home/nuevo_usuario`.

Existe otra forma manual para crear usuarios y es ejecutando *adduser*, para ello solo es necesario ejecutarlo de la siguiente forma `adduser nombre_usuario`, de esta forma la aplicación generara las líneas en el archivo */etc/passwd* y el directorio raíz del usuario.

La forma de asignarle una contraseña a los usuarios es ejecutando el comando `passwd nombre_usuario`, el sistema pedirá la nueva contraseña dos veces.

- Creación de grupos: Los grupos son importantes para delimitar los permisos entre distintos usuarios, a la vez nos permite controlar y agrupar a la variedad de tipos de usuarios que podamos tener, como ejemplo ilustrativo para el uso de grupos podemos imaginar a una empresa donde existen distintos departamentos por ende varias personas podrán tener libertad al uso de ciertas áreas pertenecientes a su departamento, a los cuales otras externas no podrán por ende siempre es mas fácil organiza por lote y luego especificar los derechos individuales para cada usuario.

Para la creación de un grupo seguimos los pasos:

- Editamos el archivo */etc/group*, en este archivo encontraremos un conjunto de líneas donde cada líneas referencia a un registro y cada línea posee el siguiente formato:

  `nombre_grupo:contraseña_grupo:ID_grupo:usuarios_grupo`.

  La identificación del grupo (GID) es aconsejable que siga la línea que ha precedido el sistema, así pues sí el último GID es 509 el nuevo deberá ser 510. Los usuarios del grupo deben ir separados por coma y pueden ser más de uno por grupo. Un usuario puede llegar a un grupo

secundario desde un grupo primario ejecutando el comando `newgroup nuevo_grupo`.

A continuación presentaremos varios comandos que nos ayudaran para la administración de cuentas en LINUX.

- Chmod: chmod es un comando donde podemos cambiar los permisos a los usarios. chmod tene a su disposicion un serie de comandos que permiten la escritura (w), lectura (r) ó ejecución (x) de un archivo. Pero también puede cambiar los permisos finales de este. Un archivo posee tres tipos de permisos, individual, al grupo y a los demás, el usuario puede definir estos permisos a través de números octales que siguen al comando chmod, el significado de los numeros octales pueden verlo en la tabla siguiente:

0	Sin permisos.
1	Ejecución.
2	Escritura.
3	Escritura y ejecución.
4	Lectura.
5	Lectura y ejecución.

6	Lectura y escritura.
7	Lectura, escritura y ejecución

Tabla 1. Permisos de archivos

Para cambiar los permisos de un archivo como *ejemplo.txt* podemos ejecutar el comando `chmod 740 ejemplo.txt` de esta manera el archivo quedará con permisos de lectura, escritura y ejecución para el dueño, lectura para su grupo y sin permisos para los demás usuarios, cabe notar que sólo el dueño ó el usuario root puede cambiar los permisos de un archivo.

- Chown: chown nos permite asignar ó cambiar propietarios a los archivos. Para ejecutar el comando sería: `chown usuario:grupo directorio/archivos`.

Como ejemplo cambiaremos de propietario el directorio */home/juancho*, directorio principal de un usuario llamado juancho a un alumno llamado enrique.

`chown enrique:alumnos /home/juancho`, con este comando el nuevo propietario será Enrique. Para asignarle todos los archivos contenido en el directorio agregamos –R luego del comando chown.

Para no arriesgar nuestro sistema ante posibles ataques es necesario conocer archivos que poseen permisos especiales como SUID y SGID, los programas con estos permisos pueden ejecutarse como un usuario distintos sin importar el usuario que lo ejecute.

Para mediar un poco de seguridad ante este problema podemos tomar estas alternativas:

- Pocos programas pueden ser SUID. Aquellos que deban serlo deben tener su propio grupo.

- Asegúrese que no puedan ejecutarse script con SUID. En caso de que los programas no necesiten imperiosamente que se defina el SUID, cambie sus permisos usando el comando, `chmod –s programa`.

1.3 INSTALACION DE LINUX

Actualmente existen muchos sistemas capaces de ofrecer servicios TCP/IP, pero existe uno que se ha convertido en el preferido por muchos usuarios y administradores debido a su flexibilidad, costo y eficiencia. Estamos hablando de

LINUX, que será nuestra plataforma de enlace para proveer los servicios que deseemos tener.

Para entender sobre la instalación de LINUX debemos tener claro los siguientes aspectos, el kernel es de distribución gratuita y de código abierto, esto quiere decir que cualquier persona puede usarlo, cambiarlo y configurarlo para su mejor acomodo, así mismo cualquier desperfecto que posea y ocurra pérdida de datos ó mal funcionamiento de su máquina nadie se hará responsable por ello.

Existen distintas distribuciones de LINUX, esto no quiere decir que sean distintos, pero si pueden variar las distribuciones que estos llevan consigo. Entre las distribuciones actuales podemos destacar Mandrake, SuSe, Red Hat, Caldera, Debian, Slackware, etc. Debemos aclarar que algunas distribuciones no soporta el manejo de paquetes Red Hat (RPM). También pueden ofrecer versiones gratuitas y otras licenciadas dependiendo del software que los proveedores hayan incluido en la distribución.

Cualquier distribución LINUX es legal mientras respete la licencia GPL (General Public License).

1.3.1 Preparativos antes de la instalación. La instalación de LINUX comienza desde el mismo momento en que el usuario desea instalarlo en su máquina, en ese momento necesita determinar si lo usará compartido con el sistema que tendría instalado, ya que LINUX acepta la vinculación con otros sistemas operativos. Luego debe decidir que distribución de LINUX utilizará dependiendo de lo que ofrezca cada una de ella.

La elección que se hizo fue la de Mandrake 8.0 con la versión del kernel 2.4.3-20, que funciona muy óptimamente, además es 100% compatible con versiones Red Hat. Además posee un gran sistema de seguridad, ofreciendo los paquetes más comunes. Mandrake posee un gran número de colaboradores haciendo que este sistema permanezca actualizado.

La página de donde se puede descargar la iso de Mandrake 8.0 es http://www.mandrake.org/, aunque puede encontrase gratuitamente en muchas otras páginas.

Linux Mandrake 8.0 requiere:

- Procesador Pentium ó compatible

- Unidad de disco compacto
- Por lo menos 800 MB de disco duro
- Tarjeta compatible VESA 2.0
- Por lo menos 32 MB de RAM (Recomienda 64MB para la instalación gráfica).

Tipos de instalación con Mandrake 8.0:

- Instalación desde la unidad de disco compacto. Se introduce el disco instalador 1 en la bandeja, saldrá en pantalla opciones para el inicio de la aplicación instaladora. Por defecto intentara instalar en modo gráfico, en caso de falla lo hara bajo texto.

- Instalación creando un disquete de arranque. Si su ordenador no inicializa desde la unidad de discos compactos, podrá crear un disquete de arranque, para esto deberá tener microsoft windows instalado, insertar el disco compacto de instalación 1, a lo cual si el auto ejecutable inicializa saldrá en pantalla opciones para crear los disquetes de arranque, en caso que este no se ejecute, podrá encontrarlo en la carpeta "dosutils", con el nombre de rawwrite.exe. Al ejecutar la aplicación elegimos como archivo de imagen el archivo "cdrom.img" que se encuentra en la carpeta "...\images" del disco instalador, luego de elegir el archivo

de imagen damos clic en write para escribir la imagen al disquete, reiniciamos nuestra máquina con la opción iniciar desde disquete de esta manera iniciara nuestra utilidad instaladora de LINUX.

- Desde un servidor de archivos NFS. De esta manera se instala en un red. Para esto primero se debe cargar la unidad de disco compacto en una máquina que admita el sistema de archivos ISO-9660 con extensiones RockRidge, y después exportar el archivo del sistema vía NFS. Necesita conocer la ruta para el archivo del sistema exportado y el número IP, ó, si tiene configurado el DNS, el nombre del sistema.

- Desde una imagen Samba. Sí los archivos se encuentran en una máquina Windows podemos acceder a ellos a través de una imagen Samba.

- Desde un servidor FTP. Se hace necesario tener un disco de arranque para comenzar la instalación, cada sitio FTP tiene un método distinto para la descarga de los instaladores, el método para la descarga generalmente se encuentra en un archivo readme que especifica la forma de descarga de la instalación. Hay que asegurarse que nuestro programa cliente FTP se encuentre en modo binario.

- Desde el mismo disco duro. En primer lugar cree un directorio con el nombre que usted desee, y copiamos todos los archivos y carpetas a la raíz de ese directorio, hacemos un disquete de arranque y conseguimos la información del disco duro.

Para conocer el diseño propio de LINUX antes de la instalación es necesario conocer la estructura de directorios que este utiliza. LINUX reconoce todo como archivo, de estos se interpretan como carpetas y estas carpetas pueden representar una carpeta de archivos ó una partición. Por ende debemos conocer la estructura de directorios a usar y las particiones que debemos crear para cada uno de ellos.

Las utilidades de partición son muy variadas pero todas llevan un estilo de manejo semejante, al inicio debemos tener un punto de montaje que servirá como nuestra partición principal que es "/".

Preguntarnos el porqué de las particiones, es sencillo y corresponde a las siguientes razones:

- Protección a los ataques de negación de servicio.

- Proteción ante los programas SUID renegados.

- Rapido inicio.

- Facilitamiento al mantenimiento y realización de copias de seguridad.

- Limita el tamaño en cada sistema de archivos para su crecimiento, sólo para que cada directorio no pueda acaparar espacio de otro en caso de que uno no lo quiera. (Es el caso de estar activo el programa LVM (Local Volume Manager) que permite cambiar esta opción).

- Habilidad para mejorar el control del sistema de archivos.

Para óptima seguridad es necesario particionar el disco y colocar los directorios en diferentes particiones, a continuación presentaremos un esquema de particiones que se podrían considerar ideal para un servidor.

Tomaremos como medida un disco duro de 3.2 GB, los valores correspondiente a cada partición se puede observar en la siguiente tabla

Partición / Directorios	***Tamaño***	***Contenido de la Partición***
/boot	5MB	Conserva las imagenes del núcleo de LINUX
/usr	512 MB (debe ser tan alto como el número de archivos binarios a instalar)	Conserva los archivos de los programas que se usan en el sistema.
/home	1140 (10 MB por cada usuario 114x10=1140)	Conserva los archivos de los usuarios.
/chroot	256 MB (debe ser lo suficientemente alto para abastecer a los programas a "enjaular")	Contiene los archivos de configuración y ejecución de programas que sean susceptibles a ataques.
/cache	256 MB	Contiene los archivos de nuestro servidor proxy.
/var	256 MB	Contiene los archivos que cambian cuando el sistema trabaja.

Partición / Directorios	***Tamaño***	***Contenido de la Partición***
<SWAP>	128 MB (será el doble para memorias RAM menores ó iguales a 64 MB e igual tamaño si es mayor a 64 MB).	Es la partición que usa el sistema como memoria virtual.
/tmp	256 MB	Guarda los ficheros en uso para el sistema y los elimina al reiniciar.
/	256 MB	Nuestro punto de montaje ósea el directorio raíz.

Tabla 2. Diseño de particiones para un LINUX seguro.

La aplicación que controla la inicialización de LINUX ó cualquier otro sistema se localiza generalmente en el master boot record de nuestra unidad ó en el primer sector de nuestra partición de inicio y podemos en ciertos casos elegir entre varias que ofrece LINUX tales como LILO (Linux Loader) ó el GRUB (Grand Unified

Bootloader), que nos permite tener varios sistemas operativos a la vez en una misma máquina y hasta en una misma unidad.

Ya particionada correctamente nuestra unidad podemos definir los filesystem de las particiones. LINUX siempre debe tener como mínimo dos tipos de estructura de directorios, una nativa y una SWAP; esta partición SWAP es obligatoria sólo para sistemas con 16 MB ó menos de RAM, el tamaño mínimo es de 16 MB y el tamaño máximo es de 1 GB. Hasta los kernel versiones 2.1.x es recomendado colocar la partición SWAP en los primeros sectores físicos de nuestra unidad. Con la variedad de filesystem del momento podemos cambiar esta estructura de nuestra partición nativa para mejorar su desempeño.

1.3.2 Comenzando la instalación. Luego de escoger la versión, verificar que la máquina contenga las especificaciones básicas y escoger el método de instalación más accesible. Empezamos la instalación de nuestro sistema LINUX. Antes de ello debemos listar los dispositivos de hardware que tengamos y veamos si nuestra versión de LINUX facilita los drivers para estos dispositivos, en caso de no encontrarla podemos buscar en Internet una lista de drivers más reciente.

Todos los LINUX compatibles con RED HAT generalmente poseen un mismo esquema de instalación, así pues describiremos la instalación generalmente con los pasos y requisitos que puedan generarse durante la instalación, debido a que utilizan la utilidad LISA que es usada para la instalación y configuración del sistema.

Como primer paso para una instalación es conocer su equipo, especificaciones de la placa madre, tarjeta gráfica, tarjeta de sonido, monitor, dispositivos scsi, teclado, ratón, impresora, puertos y demás dispositivos que puedan tener. Para esto debemos contestarnos las siguientes preguntas:

- ¿Cuántos discos duros poseemos?
- ¿De que tamaño es cada disco?
- En caso de tener varios HDD ¿cuál es el primario?
- ¿Que tipo de disco tenemos (IDE ó SCSI)?
- ¿Que tamaño de RAM tenemos?
- ¿Tenemos dispositivos SCSI?, ¿Marca y modelo?
- ¿Tenemos dispositivos RAID?, ¿Marca y modelo?
- Tipo de ratón.
- ¿Cuantos botones que posee el ratón?

- ¿Cuál es el puerto de comunicación del ratón (COM, PS2, ..)?
- ¿Cuál es la especificación de nuestra tarjeta de video?
- ¿Cuales son las especificaciones de nuestro monitor?
- ¿Cómo nos conectamos con Internet?
- Dirección IP
- Netmask (mascara de red)
- Dirección del gateway (puerta de enlace)
- Dirección de nuestro DNS.
- Nombre del servidor de dominio
- Nombre de nuestro host
- Tipo de tarjeta de red y sus respectivas especificaciones.
- Numero de tarjetas de red en uso y sus especificaciones.

El segundo paso es analizar los datos que vengan con la distribución, especificaciones, BUGS, requisitos, para saber como podría quedar instalado en nuestra máquina.

Tercer paso, como cada distribución posee un distinto programa para instalación no podemos afirmar un orden de datos a ingresar pero podemos mencionar que pedirá todo lo relacionado en esta lista:

- Lenguaje: Debemos elegir el lenguaje en que se instalará LINUX sea español, ingles, esperanto, etc.. que luego será utilizado. Es importante el lenguaje debido a que se determinará el mapa de caracteres a usar esto identificara y distribuirá los caracteres en nuestro teclado.

- Ratón: Debemos especificar las características de nuestro ratón, (Número de botones 2 ó 3), así también especificaremos el tipo de teclado que poseemos (Microsoft, Genius, ..), el puerto de comunicación (COM, PS2) y las características propias de él (óptico, inteligente, etc..).

- Teclado: Ingresamos los datos concernientes a nuestro teclado tales como numero de teclas (104), distribución ó lenguaje (español, catalán, ingles, internacional, etc..), tipo (QWERTY).

- Unidades y particiones: Se debe saber que unidades se usarán y que tamaño deberá poseer cada una, el filesystem de cada unidad deberá ser elegido en esta parte, aquí podrá ejecutarse otra aplicación que nos permitirá cambiar las particiones existentes, sabemos que LINUX debe montarse por lo mínimo en una

partición Linux nativa con un punto de montaje en “/” y al menos una partición SWAP.

- Tarjeta Gráfica: Si queremos usar el modo gráfico de LINUX debemos de especificar el modelo de nuestra tarjeta gráfica (Nvidia Gforce2 32 MB aceleración 3D, S3 Vigo++, etc..).

- Monitor: Tipo de monitor que poseemos (View Sonic pro, Samsung, Acer 4T).

- Paquetes a instalar: Se dará una lista por grupos de paquetes por tipo de utilidad que poseen y luego una individual que nos permitirá elegir los programas individualmente que vayamos a instalar.

- Conexión a Internet: Debemos proporcionar todo lo relacionado con nuestra conexión a Internet.

- Impresora: Debemos especificar el tipo de impresora que tenemos y el tipo de comunicación que tenemos con ella.

Antes de instalar usted debe tener claro el número de programas a instalar y el porqué va a instalarlo. Así pues debe tener en mente para que usara el sistema para que este instale lo que usted desea. Por ende cabe pensar en conocer los paquetes durante la instalación se podrá ver mensajes acerca de los programas ostensibles a instalar.

Luego de la instalación de los paquetes sigue siempre la configuración de nuestro entorno visual, LINUX trabaja con demonio denominado Xfree86 que nos prestará el servicio del entorno gráfico. Luego podemos elegir el gestor de ventanas a usar previamente instalado como puede ser el Gnome ó el KDE entre muchos otros más.

Luego de esta rutina de pasos la utilidad instaladora reiniciara la máquina sacará usted el disco compacto de la bandeja, de esta manera iniciará por primera vez su sistema LINUX.

Luego de la instalación es recomendable desintalar algunos programas que son instalados por defectos que pueden presentar huecos de seguridad a nuestro sistema, estos programas son:

- pump: Este software se usa si se desea adquirir una dirección IP a través de DHCP. No es necesario.
- mt-st: Este programa solo es necesario cuando hay dispositivos de cintas magnéticas en nuestro equipo.

- eject: Este programa sirve para abrir la bandeja de dispositivos de entrada y salida como floppy, cdrom, cintas magnéticas, entre otros. Solo es necesario si tenemos cintas de backup en el servidor.

- metamail: Esta aplicación que usa un archivo mailcap para mostrar los archivos adjuntos a un mail. No es necesario.

- apm: Este paquete es usado para ver el nivel de bateria para los equipos portátiles. No es necesario para un servidor.

- kernel-pcmia-cs: Este paquete controla las tarjetas pcmia en los portatiles. No es necesario para un servidor.

- linuxconf: Este paquete puede ayudarnos para la configuración del sistema, si usted es un usuario avanzado puede eliminarlo.

- getty_ps: Este paquete contiene programas que son usados para aceptar login por consola ó en una terminal del sistema. No es necesario.

- setconsole: Esta aplicación básica del sistema se encarga de configurar los archivos /etc/inittab, /dev/systty y /dev/console que son los que manejan una nueva consola. No es necesario.

- isapnptools: este paquete contiene utilidades para configurar tarjetas ISA Plug and Play. No es necesario.

- set serial: Esta utilidad es usada para configurar el puerto serial. No es necesaria.

- kudzu: Es un herramienta que verifica la disponibilidad del hardware en el sistema al inicio, esta herramienta puede verificar que hardware a sido removido ó añadido al sistema.

- raidtools: este paquete es usado para mantener el software de dispositivos RAID instalados en el sistema. Depende si utiliza ó no RAID.

- gnupg: este software remplaza al PGP y sirve para encriptar datos. Podíamos instalarlo luego ó el PGP.

- pciutils: este paquete contiene varias utilidades para inspeccionar y configurar dispositivos conectados al bus PCI. Nosotros podemos usar otros métodos.

- rmt: esta utilidad provee acceso a una red remota para hacer backup. La seguridad es puesta a riesgo desde que rmt dependa de rsh para trabajar.

1.4 CONFIGURACION Y OPTIMIZACION DE NUESTRO SISTEMA OPERATIVO LINUX.

La configuración y optimización de nuestro sistema se basa en dos aspectos, la seguridad y el desempeño que pueda tener nuestro sistema. En este apartado trataremos el diseño con la aplicación conjuntamente.

Trataremos de dar generalidades sobre la configuración y optimización del sistema sin introducirnos tanto en cada uno de los temas. Sí es un usuario avanzado podrá ver muchas cosas de lo que usted ya sabe con detalle.

Muchas distribuciones de LINUX, en su proceso de instalación suele configurar los archivos de manera excelente, por lo cual no se hace necesaria la configuración a mano.

1.4.1 Sistema general de seguridad en LINUX.

1.4.1.1 Condiciones del hardware. El hardware posee gran cantidad de configuraciones que nos permite asegurar aún mas nuestro sistema, el control de seguridad del hardware comienza desde la BIOS las contraseñas de la BIOS actualmente son muy conocidas por esta razón es necesario la actualización y cambio de estas contraseñas. No podemos basar un sistema a la seguridad de la BIOS ya que es muy débil ante cualquier ataque físico ó por software. El hardware que podamos usar en la red debe contar con ciertas normas de seguridad y usted como administrador de la red debe conocer las posibles fallas que pueda este tener, es bien conocido que muchos routers existentes manejan contraseñas por defecto lo cual ofrece al atacante un agujero de seguridad para que este pueda acceder desde el router a cualquier máquina en la red.

A pesar de que LINUX instala por defecto en las particiones el **filesystem ext2fs**, no quiere decir que sea la única opción al respecto ya que las particiones nativas

de LINUX podemos cambiarle la estructura de archivos, a otra más segura y estable como es el **REYSER FS**, este paso puede hacerlo desde la instalación al seleccionar las particiones a usar y reformateando ó cambiarla luego de la instalación del sistema, descargando el archivo de la página http://www.namesys.com/download.html y siguiendo las instrucciones que puede encontrarlas en http://www.namesys.com/install.html.

1.4.1.2 Condiciones de acceso al sistema. Podemos manejar los accesos al sistema a partir de las contraseñas, para ello debemos recomendar que estas sean mayores ó iguales a 8 digitos, no deben ser triviales, deben tener un tiempo de expiración para que sean renovadas, el login debe tener un límite de intentos. Para determinar la longitud editamos el archivo */etc/login.defs* y cambiamos la línea `PASS_MIN_LEN 5` por `PASS_MIN_LEN 8`.

Como hemos dicho la cuenta *root* es muy poderosa para ello debemos controlarla, un método consiste en controlar el período de uso de esta cuenta para ello debemos editar el archivo */etc/profile* y después de la línea que diga `HISTFILESIZE=` , añadimos `TMOUT=7200`, el valor es dado en segundos, de esta manera todos los usuarios podrán usar el sistema por lapsos de 2 (dos) horas. Sí

queremos que existan usuarios sin ese control editamos el archivo *.bashrc* y configuramos individualmente a cada usuario.

Existen consolas que pueden ejecutar comandos, y un usuario malintencionado puede usarlos para romper la seguridad del sistema por ende debemos desinstalarlos para ello debemos removerlos del directorio */etc/security* como ejemplo desinstalaremos algunos de esta manera:

```
[root@lhost] / # rm –f /etc/security/console.apps/halt
[root@lhost] / # rm –f /etc/security/console.apps/poweroff
[root@lhost] / # rm –f /etc/security/console.apps/reboot
[root@lhost] / # rm –f /etc/security/console.apps/shutdown
[root@lhost] / # rm –f /etc/security/console.apps/xserver
```

La biblioteca de funciones *Linux-PAM* es instalada por defecto en nuestro sistema esta alberga un número de utilidades para el administrador y a la vez ofrece acceso a los programas, archivos, consolas y otros dispositivos. Entonces debemos comentar sus líneas y colocarla en el directorio */etc* para que no pueda ser usada por otros usuarios, para ello podrá usar el siguiente script:

Creamos el archivo *lfuera.sh* con el comando *touch lfuera.sh* y añadimos las líneas:

```
# ! /bin/sh
cd /etc/pam.d
for i in *; do
sed ‘/[^#].*pam_console.so/s/^/#/’ <$i> foo && mv foo $I
done
```

Lo hacemos ejecutable `chmod 700 lfuera.sh` y lo ejecutamos de la siguiente forma `./lfuera.sh` de esta manera deshabilitamos la consola.

Tenemos también que configurar el archivo */etc/securetty*, este archivo especifica el terminal en que se ejecutará el usuario *root*, editamos el archivo */etc/securetty* y comentamos las líneas tal como se ve:

```
tty1
#tty2
#tty3
#tty4
#tty5
#tty6
#tty7
#tty8
```

De esta manera el usuario *root* sólo podrá usar la terminal `tty1`.

Como veníamos diciendo algunas cuentas son instaladas por defecto en nuestro sistema LINUX, algunas no son necesarias y podemos eliminarlas para evitar algún posible ataque. Aquí damos un pequeño ejemplo de la eliminación de cuentas en un sistema:

```
[root@lhost] / # userdel adm
[root@lhost] / # userdel lp
[root@lhost] / # userdel sync
[root@lhost] / # userdel shutdown
[root@lhost] / # userdel halt
[root@lhost] / # userdel news
[root@lhost] / # userdel uucp
[root@lhost] / # userdel operator
[root@lhost] / # userdel games
[root@lhost] / # userdel gopher
[root@lhost] / # userdel ftp
```

El comando userdel sólo eliminará la entrada de la cuenta, los directorios de la cuenta se conservarán, sólo usted puede eliminarlos. Luego eliminaremos los grupos que llevan estas cuentas también:

```
[root@lhost] / # groupdel adm
[root@lhost] / # groupdel lp
[root@lhost] / # groupdel dip
[root@lhost] / # groupdel pppusers
[root@lhost] / # groupdel popusers
[root@lhost] / # groupdel news
[root@lhost] / # groupdel uucp
[root@lhost] / # groupdel slipusers
[root@lhost] / # groupdel games
```

Inmunizamos todos nuestros archivos de cuentas, con los siguientes comandos:

```
[root@lhost] / # chattr  +i /etc/passwd
[root@lhost] / # chattr  +i /etc/shadow
[root@lhost] / # chattr  +i /etc/group
[root@lhost] / # chattr  +i /etc/gshadow
```

Para restringir la cuenta *root* debemos restringir el acceso que puedan tener los usuarios para ejecutar comandos ó programas con esta cuenta, para ello debemos bloquear el acceso al comando `su`, que nos permite ejecutar como superusuario conociendo la contraseña del *root*. Para bloquear este comando de los usuarios normales, debemos editar el archivo */etc/pam.d/su* y añadir las siguientes líneas:

```
auth sufficient /lib/security/pam_rootok.so debug
auth required /lib/security/pam_wheel.so group=wheel
auth required /lib/security/pam_pwdb.so shadow nullok
account required /lib/security/pam_pwdb.so
password required /lib/security/pam_cracklib.so
password required /lib/security/pam_pwdb.so shadow
use_authtok nullok
session required /lib/security/pam_pwdb.so
session optional /lib/security/pam_xauth.so
```

En caso de asignar el comando su a algún usuario en especial, se debe ejecutar el siguiente comando desde el shell, `usermod –G# n_usuario`, donde `G` es una lista de grupos suplementario, `#` es el ID numérico del usuario y `n_usuario` es el nombre del usuario.

Como medida paranoica debemos también aplicar límites a la disponibilidad de los recursos en el sistema para ello debemos editar el archivo */etc/security/limits.conf* y añadimos ó cambiamos las líneas:

```
*hard    core 0
*hard rss 5000
*hard nproc 20
```

Donde * significa a todos los usuarios en el sistema (exceptuando el *root*), `core 0` prohíbe la creación de archivos en el núcleo, `rss 5000` sólo se le permitirá el uso de 5 MB en memoria, `nproc 20` sólo podrá ejecutar 20 procesos a la vez.

Luego editamos el archivo */etc/pam.d/login* y añadimos al final la línea :

```
session required /lib/security/pam_limits.so
```

A lo cual nuestro archivo podrá quedar de esta forma:

```
#%PAM-1.0
```

```
auth required /lib/security/pam_securetty.so
auth required /lib/security/pam_pwdb.so shadow nullok
auth required /lib/security/pam_nologin.so
account required /lib/security/pam_pwdb.so
password required /lib/security/pam_cracklib.so
password required /lib/security/pam_pwdb.so nullok
use_authtok md5 shadow
session required /lib/security/pam_pwdb.so
session required /lib/security/pam_limits.so
#session optional /lib/security/pam_console.so
```

Finalmente cambiamos la línea `ulimit -c 1000000` del archivo */etc/profile* por `ulimit -S -c 1000000 > /dev/null 2<&1`.

El punto a tratar ahora es muy importante ya que en este punto aprenderemos a designarle las propiedades a las particiones para que cada una de estas pueda tener un nivel de seguridad distinta a las otras.

Para cambiar las opciones de seguridad en cada partición debemos editar el archivo /etc/fstab, cada partición puede tener una ó más propiedades tal como lo mencionaremos:

- defaults: permite todo quota, suid, lectura y escritura.
- noquota: no permite la configuración de quota para usuarios en la partición.
- nosuid: no permite el uso de SUID ó SGID por los usuarios en esta partición.
- nodev: no permite el acceso especial a dispositivos desde la partición.
- noexec: no permite la ejecución ningún binario en esta partición.
- quota: permite el uso de la quota a los usuarios.
- ro: sólo lectura esta partición.
- rw: activa lectura-escritura en esta partición.
- suid: activa el uso de los comandos SUID ó SGID en esta partición.

Ahora al editar el archivo */etc/fstab* podemos ver las líneas de las partciciones que cambiaremos más ó menos de esta manera:

```
/dev/sda11 /tmp ext2 defaults 1 2
/dev/sda6 /home ext2 defaults 1 2
```

Donde primero nos muestra el dispositivo físico, luego el punto de montaje, sigue el tipo de sistema de archivo que utiliza, luego las opciones. Cambiaremos las opciones de estos directorios de la siguiente manera:

```
/dev/sda11 /tmp ext2 defaults,rw,nosuid,nodev,noexec 1 2
```

```
/dev/sda6 /home ext2 defaults,rw,nosuid,nodev 1 2
```

Luego debemos remontar las particiones reconfiguradas de la siguiente manera:

```
[root@lhost] /#mount –oremount /home/
[root@lhost] /#mount –oremount /tmp/
```

Al mostrarnos paranoicos con la seguridad del sistema debemos mover los comandos RPM a un disco distinto para que cualquier atacante no pueda instalar, basándose en este programa troyanos ó paquetes que puedan afectar la seguridad de nuestro sistema, para ello debemos seguir los siguientes pasos:

```
[root@lhost] /# mount /dev/fd0 /mnt/floppy/
[root@lhost] /# mv /bin/rpm /mnt/floppy/
[root@lhost] /# umount /mnt/floppy
```

No debemos desinstalar el programa RPM completamente porque de esa manera no podríamos reinstalarlo ú otro software que necesite comandos RPM. En caso que no quiera mover los archivos de este programa puede cambiar los permisos de la siguiente manera:

```
[root@lhost] /# chmod 700 /bin/rpm
```

De esta forma sólo el usuario root podrá acceder a este programa.

Hay que saber que LINUX guarda para cada usuario un historial de comandos que el usuario puede volver a utilizar este se guarda en el archivo *~/.bash_history* donde ~/ corresponde al directorio *home* de cada usuario, para evitar que este historial sea muy extenso debemos editar el archivo */etc/profile* y cambiar en las líneas *HISTFILESIZE, HISTSIZE* la cantidad de comandos que se deban guardar en el sistema.

Para evitar que crackers puedan ejecutar comandos por otras cuentas es necesario borrar el archivo de historial editando el archivo */etc/skel/.bash_logout* y añadimos la siguiente línea:

```
rm -f $HOME/.bash_history
```

Como ya hemos hablado un mecanismo de protección es asegurarnos desde el inicio y cabe la duda que el LILO pueda ser nuestro enemigo intimo en relación a la seguridad. Generalmente LILO puede tener un archivo de configuración por

defecto a lo cual hay que añadir líneas para que este funcione en una forma más segura, las líneas a añadir pueden ser:

- timeout =00, esta opción debe ser puesta en 0 sí solamente tenemos LINUX en nuestra máquina.

- restricted, esta opción pregunta por una contraseña para inicializar el sistema operativo.

- passwd= < contraseña >. En este campo se encuentra la contraseña.

Sí editamos el archivo de configuración del LILO, ubicado en */etc/lilo.conf*, podemos observarlo más ó menos de esta manera:

```
boot=/dev/sda.44
map=/boot/map
install=/boot/boot.b
prompt
timeout=00
Default=linux
restricted
```

```
password=<password>
image=/boot/vmlinuz-2.2.12-20
label=linux
initrd=/boot/initrd-2.2.12-20.img
root=/dev/sda6
read-only
```

Podemos observar que sólo se usa LINUX en este sistema. Luego debemos cambiar los permisos de este archivo ejecutando:

```
[root@lhost] / #chmod 600 /etc/lilo.conf
```

Así sólo el usuario *root* podrá leer solamente el archivo. Luego para más seguridad agregamos la opción de inmutable para el archivo de la siguiente manera:

```
[root@lhost] / #chattr +i /etc/lilo.conf
```

Para una futura actualización de este archivo debemos cambiarle los atributos y permisos.

Debemos quitar la opción de CTRL+ALT+DEL para el apagado del sistema editando el archivo /etc/inittab comentando la línea:

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

como:

```
#ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

Para que esto tenga efecto debemos ejecutar el comando:

```
[root@lhost] /#/sbin/init q
```

Para agregar seguridad al sistema debemos cuidar que los procesos que se ejecutan normalmente no puedan parar ú otros a ejecutarse por control de otros usuarios, para ello debemos cambiar los permisos de los archivos que residen en ese directorio, de la forma:

```
[root@lhost] /# chmod -R 700 /etc/rc.d/init.d/*
```

No hay que olvidarse que al instalar un nuevo programa cambiar los archivos que este programa instale en el directorio.

Hablando del directorio */etc/rc.d*, sabemos que LINUX provee en cada login cierta información que pueda ser usada por el atacante para invadir nuestro sistema, para ello hay que editar el archivo /etc/rc.d/rc.local y comentar las líneas tal como se muestra:

```
# Esto se sobrescribirá en cada inicialización /etc/issue.
#puede ser cambiado
#echo "" > /etc/issue
#echo "$R" >> /etc/issue
#echo "Kernel $(uname -r) on $a $(uname -m)" >> /etc/issue
#
#cp -f /etc/issue /etc/issue.net
#echo >> /etc/issue
```

Y removemos los archivos:

```
[root@lhost] /# rm -f /etc/issue
[root@lhost] /# rm -f /etc/issue.net
```

De esta manera la información de nuestro sistema no se mostrará por ningún login sin importar de donde venga.

Para el administrador es importante mantener fuera del alcance de cualquier usuario programas que tengan activo el bit SUID ó SGID, es tarea del administrador conocer que programas contienen este bit activo y se encuentren instalados en el sistema, para encontrar estos programas puede ejecutar el comando:

```
[root@lhost] / #find /  -type f  \(  -perm -04000 -o -perm -
02000)/ \-exec ls 'lg {} \;
```

De este comando aparecerá una lista con nombres y ubicación de archivos conesos permisos. Para cambiar los permisos de estos archivos debemos ejecutar el comando:

```
[root@lhost] /# chmod a-s ubicación_archivo
```

Usualmente los intrusos suelen dejar archivos en nuestro servidor, estos archivos pueden ocultarlos de manera propia para que el usuario *root* no pueda observarlos con un sencillo ls, por ende es necesario que el administrador conozca los

métodos y la manera de buscarlos para evitar que colmen el disco duro de información oculta. Las maneras más sencillas de ocultar un archivo en LINUX es anteponiendo un punto ó anteponiendo `..^G` (nó Ctrl `G`), para observar los archivos ocultos de esta manera podemos ejecutar el comando:

```
[root@lhost] / # find / -name ".. " -print -xdev
[root@lhost] / # find / -name ".*" -print cat –v.46
```

1.4.1.3 Condiciones de acceso por la red. Cómo sabemos existe un super-servidor llamado inetd este provee muchos servicios en un servidor LINUX, muchos de manera insegura para ello debemos configurar este servidor para evitar ataques por la red con este servidor.

El archivo de configuración de los servicios es el archivo */etc/inetd.conf*, editamos este archivo y deshabilitamos servicios que no usaremos comentando las líneas.

Cambiamos los permisos en el archivo de manera:

```
[root@lhost] / #chmod 600 /etc/inetd.conf
```

Nos aseguramos que *root* es el propietario del archivo.

```
[root@lhost] /# stat /etc/inetd.conf
File: "/etc/inetd.conf"
Size: 2869 Filetype: Regular File
Mode: (0600/-rw-------) Uid: ( 0/ root) Gid: ( 0/ root)
Device: 8,6 Inode: 18219 Links: 1
Access:WedSep2216:24:162001(00000.00:10:44)
Modify:MonSep2010:22:442001(00002.06:12:16)
Change:MonSep2010:22:442001(00002.06:12:16)
```

Editamos el archivo */etc/inetd.conf* y deshabilitamos los servicios ftp, telnet, shell, login, exec, talk, ntalk, imap, pop-2, pop-3, finger, auth, etc. Así como todos los que no planeamos usar. El archivo podrá quedar de la siguiente manera:

```
# Debe desactivar el servicio para observar los cambios
'killall -HUP inetd'
#
#echostreamtcpnowaitrootinternal
#echodgramudpwaitrootinternal
#discardstreamtcpnowaitrootinternal
```

```
#discarddgramudpwaitrootinternal
#daytimestreamtcpnowaitrootinternal
#daytimedgramudpwaitrootinternal
#chargenstreamtcpnowaitrootinternal
#chargendgramudpwaitrootinternal
#timestreamtcpnowaitrootinternal
#timedgramudpwaitrootinternal
#
# Estos servicios son estándar.
#
#ftpstreamtcpnowaitroot/usr/sbin/tcpdin.ftpd -l -a
#telnet streamtcpnowaitroot/usr/sbin/tcpdin.telnetd
#
# Shell, login, exec, comsat and talk are BSD protocols.
#
#shellstreamtcpnowaitroot/usr/sbin/tcpdin.rshd
#loginstreamtcpnowaitroot/usr/sbin/tcpdin.rlogind
#execstreamtcpnowaitroot/usr/sbin/tcpdin.rexecd
#comsatdgramudpwaitroot/usr/sbin/tcpdin.comsat.47
#talkdgramudpwaitroot/usr/sbin/tcpdin.talkd
#ntalkdgramudpwaitroot/usr/sbin/tcpdin.ntalkd
```

```
#dtalk streamtcpwaitnobody/usr/sbin/tcpdin.dtalkd
#
# servicios pop e imap para mail
#
#pop-2streamtcpnowaitroot/usr/sbin/tcpdipop2d
#pop-3streamtcpnowaitroot/usr/sbin/tcpdipop3d
#imapstreamtcpnowaitroot/usr/sbin/tcpdimapd
#
# El servicio uucp para internet.
#
#uucpstreamtcpnowaituucp/usr/sbin/tcpd /usr/lib/uucp/uucico -
l
#
#
# Esto puede ser que no lo necesite
#
#tftpdgramudpwaitroot/usr/sbin/tcpdin.tftpd
#bootpsdgramudpwaitroot/usr/sbin/tcpdbootpd
#
#
#El servicio Finger ofrece mucha información
```

```
#mejor cerremoslo
#fingerstreamtcpnowaitroot/usr/sbin/tcpdin.fingerd
#cfingerstreamtcpnowaitroot/usr/sbin/tcpdin.cfingerd
#systatstreamtcpnowaitguest/usr/sbin/tcpd/bin/ps -auwwx
#netstatstreamtcpnowaitguest/usr/sbin/tcpd/bin/netstat     -f
inet
#
# Authentication
#
#authstreamtcpnowaitnobody/usr/sbin/in.identd in.identd -l -e
-o
#
# End of inetd.conf
```

Luego paramos el proceso demonio del servicio:

```
[root@lhost] /# killall -HUP inetd
```

Una medida más de seguridad es colocar el archivo inmutable, de la siguiente manera:

```
[root@lhost] /# chattr +i /etc/inetd.conf
```

Este comando prevendrá de cualquier cambio accidental que tenga el archivo. Los *TCP_WRAPPERS* son usados para incrementar la seguridad en un sistema, este mecanismo es usado comunmente por los servicios en LINUX, los *TCP_WRAPPERS* trabajan en función de dos archivos:

```
/etc/host.deny
/etc/host.allow
```

Para asegurarnos que ningún servidor entre al de nosotros sin una autorización, añadimos las líneas:

```
# Deny access to everyone.
ALL: ALL@ALL, PARANOID
```

Donde el parámetro `PARANOID`, hace que las conexiones ftp y telnet demoren el tiempo suficiente mientras el servidor realiza la operación *DNS_lookup* por tiempo máximo de espera.

En el archivo *host.allow*, indicamos que máquinas y servicios a los que pueden acceder. Como ejemplo una línea `sshd: 208.164.186.1 gate.cutb.edu.co` donde `sshd` es el servicio de acceso remoto a un shell seguro, `208.164.186.1` es la dirección IP del cliente y `gate.cutb.edu.co` es el servidor que usa el cliente para comunicarse.

Sí no queremos que se muestre los *TCP_WRAPPERS* configurados en los servicios editamos el archivo */etc/inetd.conf* y agregamos –h al final de la línea de el servicio que no queramos que muestre la información acerca de este servicio. Por ejemplo para el servicio telnet:

```
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd –h
```

LINUX utiliza una librería para resolver los nombres de dominio esto lo hace por medio del archivo */etc/host.conf*, las entradas a este archivo indican los servicios usados y el orden en que se emplean, sí queremos configurar de manera segura este archivo debemos añadir las líneas:

```
# Mira los nombres de los DNS antes del archivo /etc/host.
order bind,hosts
```

```
# Tenemos máquinas con más de una IP.
multi on
# Prueba el spoof para cada IP.
nospoof on
```

Los servicios que se ofrece LINUX los coloca en el archivo */etc/services*, para evitar ataques debemos inmunizar este archivo para prevenir accidentes activando el siguiente comando desde el shell, *chattr +i /etc/services.*

Los ataques a través de contestar peticiones por medio de ping son más constantes de lo que se cree, servidores como yahoo.com y altavista.com han caído por medio de este ataque para asegurarnos que no haya este ataque debemos hacer que el servidor niegue cualquier petición por medio de ping para ello basta con ejecutar el comando:

```
[root@lhost]/# echo 1 > \
/proc/sys/net/ipv6/icmp_echo_ignore_all
```

Para rehabilitar hay que ejecutar el comando:

```
[root@lhost] /#echo 0 > \
```

```
/proc/sys/net/ipv6/icmp_echo_ignore_all
```

Pero esta deshabiliación solo ocurre al nosotros ejecutar el comando, sí queremos que esto ocurra automáticamente ingresamos el comando en nuestro archivo */etc/rc.d/rc.local*.

2. USO DE LOS SERVICIOS TCP/IP

Hoy en día se hace imposible hablar de TCP/IP sin nombrar Internet y viceversa. TCP/IP es el nombre de un conjunto de muchos protocolos, es decir de muchas reglas definidas con el fin de posibilitar la comunicación entre máquinas haciendo que todas hablen el mismo lenguaje, donde los más representativos son estos dos, de ahí que se le ha dicho al conjunto de este nombre.

TCP/IP nace de la necesidad de la existencia de unos protocolos de comunicaciones robustos, capaces de unir ordenadores diferentes y que pudieran reestablecerse con relativa facilidad en el caso de que un ataque destruyera parte de la red de comunicaciones.

Fue desarrollado por un equipo de investigadores del Departamento de Defensa de los Estados Unidos a partir de un proyecto llamado DARPA (Defense Advanced Research Proyects Agency), a partir de 1968. Un año después la red ya estaba operativa con el nombre de ARPANET.

Paralelamente estudiantes de universidades trabajaban en un modelo de comunicaciones parecido al ARPANET, luego todas estas entidades asociadas

con desarrolladores electrónicos y empresas se agruparon para el desarrollo integral de la red, de esta manera ARPANET se separa en dos en 1983 ARPANET para el público y MILNET para los militares.

El protocolo TCP/IP hace su aparición hasta 1974, poco a poco se fueron uniendo a otras redes y fue creciendo la red. Con la desaparición del comunismo a nivel mundial y la recesión de la guerra fría la red de redes se incorpora a nivel mundial dando su salida en la década de los ’90 (noventas).

Todas las especificaciones técnicas de este protocolo se agrupan en unos documentos llamados RFC (Request For Comments) petición por comentarios. Estos documentos están a disposición de todo el mundo e igualmente todo el mundo puede enviar datos que ayuden a mejorar la calidad de dichos protocolos. Cuando envía modificaciones a dichos protocolos, dichas modificaciones deben pasar a manos de un comité hasta llegar a su aprobación ó rechazo.

En la actualidad se trabaja sobre una nueva versión de TCP/IP conocida como IPv6 ó IPng. Cada uno de estos protocolos pertenecen a un nivel definido por el modelo OSI (Open System Interconnection) de la ISO (International Organization

for Standardizacion). El protocolo TCP pertenece al nivel de transporte (cuarto nivel del modelo) y el protocolo IP a nivel de red (nivel tres del modelo).

Uno de los principales motivos que han hecho que estos protocolos salten a la fama es, sin duda alguna la capacidad que ofrecen para independizar la comunicación entre máquinas no sólo al nivel de arquitectura. Con TCP/IP se pueden comunicar todas aquellas máquinas que posean dichos protocolos, independiente de cómo funcionen éstas.

La arquitectura TCP/IP define 4 (cuatro) niveles operativos:

- Nivel físico que es el encargado de establecer las normas de acceso a las redes físicas (a través de Ethernet, X.25, ATM, Frame Relay, SLIP-PPP, etc.).

- El nivel de red, en el que se encuentra el protocolo IP (Internet Protocol) que define las direcciones de las máquinas que forman parte de la red, la estructura de un datagrama IP lo puede observar en la figura siguiente.

Figura 3. Datagrama IP

- El nivel de transporte, en que se pueden encontrar dos protocolos diferentes, por un lado TCP (Transmision Control Protocol en la figura a seguir podemos ver el formato de este paquete), protocolo orientado a conexión y fiable, y por el otro lado UDP (User Datagram Protocol), protocolo no orientado a conexión.

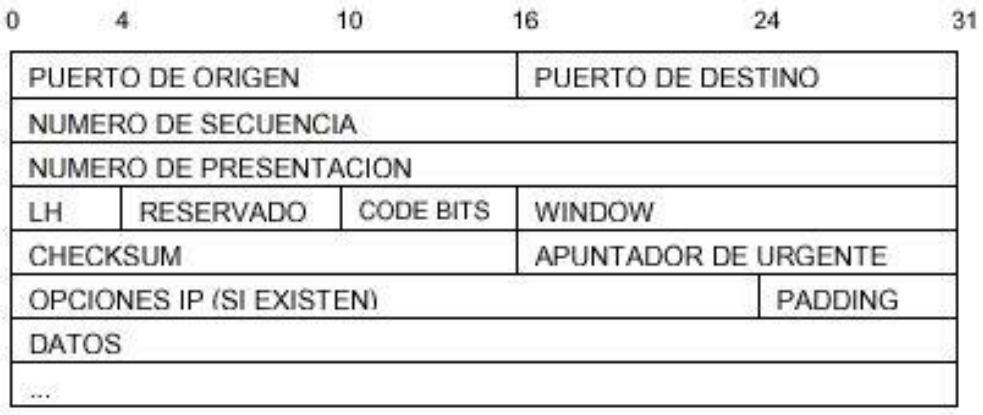

Figura 4. Datagrama UDP

- En el último nivel se encuentran los protocolos como Telnet, FTP (Protocolo de transferencia de ficheros), SMTP (Protocolo básico de transferencia de correo), HTTP, DNS (Servicio de Nombres de Dominio), SNMP (Protocolo básico de gestión de red), POP3 (Post Office Protocol 3).

La figura siguiente muestra un esquema de esta arquitectura.

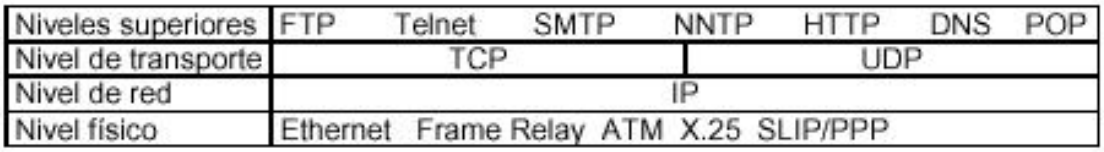

Figura 5. Arquitectura TCP

En la figura del datagrama IP pudimos observar la cabecera de un datagrama IPv4, actualmente se trabaja con IPv6 ó IPng (IP new generation). IPv6 es la modificación de la versión 4 que tiene como modificaciones importantes:

- Mayor espacio de direccionamiento: el nuevo tamaño de direccionamiento es de 128 bits con máscaras de 128 bits, (recuerde que IPv4 tiene direcciones de 32 bits). Se hace tan grande para que no haya problema en el futuro.

- Formato flexible de las cabeceras: IPv6 utiliza un formato de datagrama diferente; no define campos de longitud fija y en posiciones exactas en la cabecera, sino que usa conjuntos opcionales de cabeceras.

- Opciones mejoradas: Los datagramas incluyen información opcional de control; esto permite mejorar la gestión de los servicios y la definición de otros nuevos.

- Soporte para asignación de recursos: IPv6 reemplaza la anterior filosofía de asignación de recursos, con un mecanismo que permite la reserva de ancho de banda y recursos de transmisión. En concreto, se soportan aplicaciones que exigen ancho de banda fijo y retardo constante.

- Previsión para ampliaciones futuras: Se trata de un protocolo con una arquitectura abierta, preparado para soportar ampliaciones futuras de forma consistente y para evolucionar y adaptarse a las nuevas tecnologías en redes de transmisión ó nuevas aplicaciones.

El formato de esta versión también cambia, teniendo cabecera fija de 40 bytes y cabeceras opcionales con longitud fija de 40 bytes, las cabeceras opcionales pueden ser:

- Cabecera de opciones salto a salto.
- Cabecera de opciones de destino.
- Cabecera de encaminamiento.
- Cabecera de fragmento.
- Cabecera de autenticación.
- Cabecera de datos de seguridad de encapsulación.
- Cabeceras de niveles superiores.

El alineamiento cambia de múltiplos de 32 bits a múltiplos de 64 bits. El campo de longitud de cabecera se elimina y el campo de longitud del datagrama se sustituye por un campo de longitud de “carga de tránsito”. El tamaño de direcciones de origen y direcciones de destino aumenta en 16 bytes.

En resumen un datagrama IPv6 comienza con un encabezado base de 40 bytes que incluye campos para las direcciones de fuente y destino, el límite máximo de saltos, la etiqueta de flujo y el tipo próximo encabezado.

Así un datagrama IPv6 debe contener, cuando menos, 40 bytes además de los datos. El porqué de esta breve explicación es porque aún existen en el mundo

servidores y máquinas que funcionan con IPv4, sí desea más información puede leer el *RFC 1933: "Mecanismos de transmisión para routers y servidores IPv6"*.

2.1 DISEÑO DE UNA RED DE SERVICIOS TCP/IP.

Antes que nada debemos saber que tan grande será la red, número de usuarios, dependencias, uso de la red, acceso a otras redes, número de subredes y plataforma física.

A partir de estos datos el analista puede sacar conclusiones de el tipo de topología a usar, acceso a la red, hardware a utilizar, número de servidores, centros ó nodos de información, nivel de seguridad y servicios a instalar.

Ya que muchas de estas especificaciones nos apartan de nuestro tema principal las veremos brevemente:

- Topología de red: Existen diferentes topologías de red y pueden usarse individualmente ó en conjunto las más usadas son: bus, estrella y anillo.

- Acceso a la red: Debemos delimitar que tipo de acceso tendrán los usuarios a la red y como lo harán, es decir que hardware podrán usar para el acceso a la red, el cableado y dispositivos.

- Hardware a utilizar: El administrador debe determinar que hardware ó máquinas usará, en la red debe especificar todas.

- Número de servidores: De acuerdo a los servicios, cantidad de información, número de usuario, nivel de seguridad y estructura que le quiera dar a la red el administrador debe determinar la cantidad de servidores que implementará en la red.

- Centros ó nodos de información: El administrador debe determinar ¿cuántos centros ó nodos debe tener la red?, y el soporte para cada uno de ellos.

- Nivel de seguridad: De acuerdo a la información tratada, cantidad de usuarios, permisos de estos, reglas de la empresa y servicios a prestar, el administrador debe determinar políticas y reglas que se deben cumplir a la vez paralelamente debe existir un grupo de auditoría si se requiere que haga evaluaciones constantes del sistema montado.

- Servicios a instalar: De acuerdo al uso que se le vaya a dar al sistema, el nivel de seguridad y los requerimientos para los usuarios, el analista debe detallar los servicios a montar y el nivel de seguridad de cada uno.

Aparte de todo esto el administrador debe verificar si la red estará comunicada ó no con la Internet, para esto debe especificar el diseño para cada servicio con tal que exista integración ó no con la red de redes, en caso de que la red no posea esta cualidad debe disponer sus servicios con IP privadas, de la clase que más se adecue:

- tipo A: 10.0.0.0
- tipo B: 172.16.0.0 hasta 172.31.255.255
- tipo C: 192.168.0.0 hasta 192.168.255.255

Sí la red deberá estar conectada con la Internet, antes de tramitar nuestro dominio debemos de saber que tipo de dirección IP se almodará más a nuestras necesidades, existen tres tipos de direcciones IP tal como lo vemos en la figura siguiente.

Tipo	Nodos Disponibles	Bits Iniciales	Dirección de Inicio
A	2^{24}=167772	0xxx	0-127
B	2^{16}=65536	10xx	128-191
C	2^{8}=256	110x	192-223
D		1110	224-239
E		1111	240-255

Figura 6. Tipo de direcciones

Las direcciones tipo A son usadas para redes de gran tamaño ó para conjunto de redes asociadas. Todas las instituciones educativas están agrupadas bajo una dirección de tipo A. Las direcciones tipo B se utilizan para redes de gran tamaño con más de 256 nodos (pero con menos de 65536). Las direcciones tipo C son las que utilizan la mayoría de organizaciones. Es aconsejable que una organización disponga de varias direcciones de tipo C puesto que el número de direcciones tipo B es limitado. El tipo D se reserva para los mensajes de transmisión múltiple en la red, mientras que el tipo E se reserva para la experimentación y el desarrollo.

Para la obtención de direcciones IP válidas ó legales (son las que pueden verse ó acceder desde Internet), debe comunicarse con el centro de información de la red (NIC) por medio de los datos:

Network Solutions

ATTN: InterNIC Registration Services

505 Huntmar Park Drive

Herndon, VA 22070

Estados Unidos

(703) 742-4777

ó por Internet http://www.internic.net

En la mayoría de los casos cuando se conecta una computadora a la Internet, el proveedor del servicio de Internet es quien se encarga de registrar la dirección IP para la red.

Usted como administrador de red debe enterarse de los documentos RFC que se relacionen con los servicios ó redes que esté usando para ello puede descargarlo desde el servidor http://ftp.internic.net/rfc, como mínimo debe tener para su documentación los siguientes RFC:

- RFC791.txt: Protocolo Internet Especificación de mensajes de programa Internet DARPA.

- RFC792.txt: Protocolo de mensaje de control Internet.

- RFC793.txt: Protocolo de control de transmisión. Especificación de mensaje de programa Internet DARPA.
- RFC950.txt: Procedimiento estándar de subred Internet.

- RFC1058.txt: Protocolo de información de enrutamiento.

- RFC1178.txt: Selección de un nombre para la computadora.

- RFC1180.txt: Un tutorial de TCP/IP.

- RFC1208.txt: Un glosario de términos de trabajo en red.

- RFC1219.txt: Información sobre la asignación de números de subred.

- RFC1234.txt: Direccionamiento de tráfico IPX a través de redes IP.

Para la designación de nombres en la red, el administrador debe ser coherente en su política, los nombres asignados deben de ser sencillos y deben especificar el

nombre de la máquina de acuerdo a sus características, debe además ser un nombre sencillo y el nombre no debe contener caracteres especiales.

Con respecto a la designación de nuestro servidor, el NIC posee un árbol de designación en el cual las organizaciones de una misma índole deben estar en la misma ramificación del árbol, los nombres más comunes del árbol son:

- edu: Instituciones educativas.
- com: Comercial.
- gov: Instituciones gubernamentales.
- mil: Instituciones militares.
- net: Gestión y administración de la red Internet.
- org: otro tipo de organizaciones (generalmente sin ánimo de lucro).

El establecimiento de subredes consiste en dividir una gran red lógica en redes físicas más pequeñas.

Las razones que llevan a dividir una red van desde las limitaciones eléctricas en la tecnología de trabajo en red, hasta el mero deseo por simplificar las cosas, ubicando una red distinta en cada planta de un edificio (ó en cada departamento ó

para cada aplicación), pasando por la necesidad de disponer de ubicaciones remotas conectadas por una línea de alta velocidad.

Las redes resultantes son partes más pequeñas de la red completa y su gestión resulta más sencilla. Las subredes más pequeñas se comunican entre sí mediante pasarelas y enrutadores. Además, dentro de una organización, puede haber varias subredes que se encuentren físicamente en una misma red. De esta forma, podrían dividirse lógicamente las funciones de red en grupos de trabajo. Las subredes individuales son una división de la red completa.

Supongamos que una red tipo B se divide en 64 subredes distintas. Para ello, la dirección IP se visualiza en dos partes: red y sistema principal como la figura siguiente.

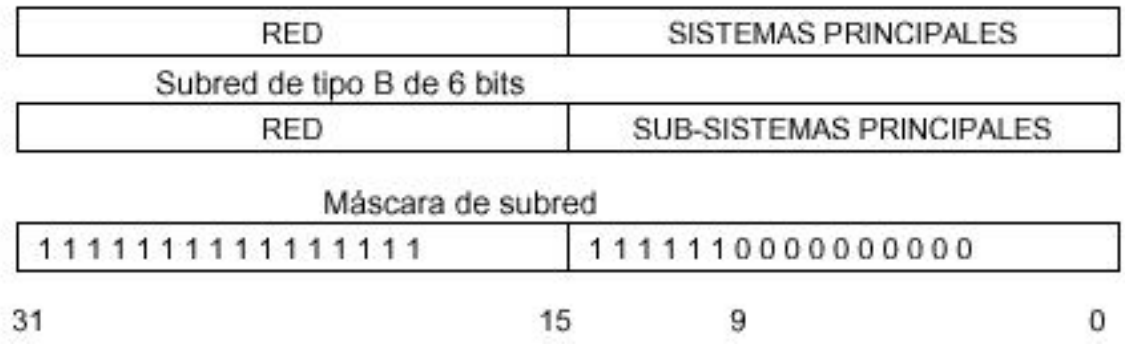

Figura7. Esquema de una dirección IP

La parte de la red se convierte en la dirección IP asignada y en los bits de información de la subred.

Básicamente, estos bits se eliminan de la parte del sistema principal. El número asignado de bits para una red de tipo B es de 16. La parte de la subred añade 6 bits, teniendo así un total de 22 bits para distinguir la subred. La división da por tanto como resultado 64 redes con 1024 nodos en cada una de ellas.

La parte de la red puede ser mayor ó menor, dependiendo del número de redes que se desee tener o del número de nodos por red.

Para establecer una red de subred debe determinarse antes donde termina la dirección de la red y dónde comienza la dirección del sistema principal. La máscara de la subred contiene todos los 1 (unos) en el campo de red y los 0 (ceros) en el campo del sistema principal.

El enrutamiento es un método para transferir información entre redes. Un enrutador funciona en la capa de red de los protocolos de red. Existen varias formas distintas de enrutar datos. La forma implantada por una red Internet es el protocolo de enrutamiento de información (RIP). El RIP ha sido diseñado para ser

utilizado en redes pequeñas y medianas y está basado en los protocolos de enrutamiento de Xerox Network System (XNS).

RIP determina la ruta del mensaje utilizando un algoritmo de enrutado de distancia vector. Este algoritmo supone que a cada ruta se le asigna un coste. Este coste puede representar el rendimiento de una red, el tipo de línea utilizando o la conveniencia de la ruta que suponga el menor coste para la transmisión del mensaje. Para mantener una lista de saltos ó nodos adyacentes el enrutador RIP debe mantener una tabla de encaminamiento que carga en memoria, esta tabla se actualiza cada 30 segundos con los enrutadores adyacentes. Cada enrutador tiene un coste máximo de 16, en caso de que este coste se sobrepase el sistema lo considera inalcanzable, así que volverá a hacer un enrutamiento por un coste menos alto. La ruta se elimina de memoria si el enrutador no la actualiza por 180 segundos.

Para redes segmentadas por x ó y motivo debe hacerse antes un análisis para observar que dispositivo de unión es mejor, para redes muy segmentadas es mejor usar puentes ya que el enrutador provoca congestión y baja el rendimiento de la red un puente no enruta ni hace traducciones en los paquetes, el enrutador es recomendados para redes que no compartan información muy a menudo.

El diseño de una red Internet es muy similar al diseño de cualquier red informática. Comprende muchos tipos de nodos, incluyendo estaciones de trabajo, servidores, impresoras, grandes computadoras, enrutadores, puentes, pasarelas, servidores de impresión y terminales. La red Internet requiere que cada uno de los dispositivos tenga su dirección IP exclusiva. Un dispositivo puede tener más de una dirección, dependiendo de su función, pero es necesaria al menos una dirección para comunicarse con el resto de los dispositivos.

Una red TCP/IP puede estar formada por varios sistemas conectados a una red de área local o por cientos de sistemas conectados a miles de sistemas en Internet. Cada organización puede crear el tipo de red que mejor se ajuste a sus necesidades. En la figura a continuación podemos observar una red sencilla compuesta por un servidor y varias estaciones de trabajo.

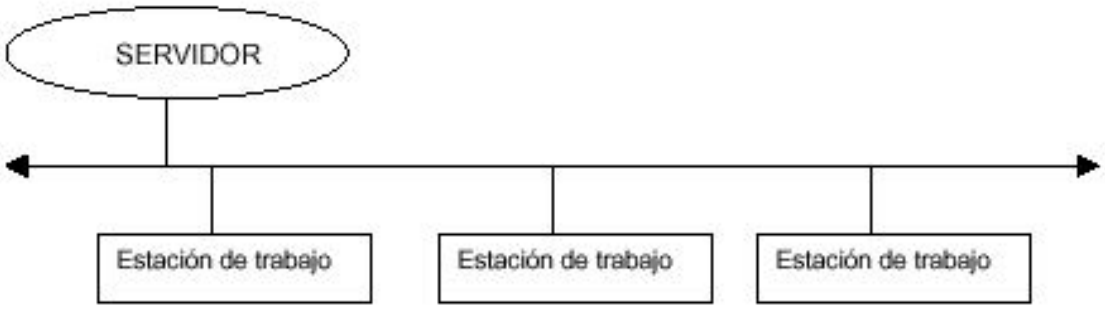

Figura 8. Red sencilla

En cambio en la figura siguiente podemos observar una red más compleja compuesta de tres redes independientes que están interconectadas mediante una combinación de enrutadores y servidores. Cada una de las estaciones de trabajo y computadoras de cada segmento pueden estar aisladas ó no, en cuanto a la utilización de la utilización en una de las dos redes restantes.

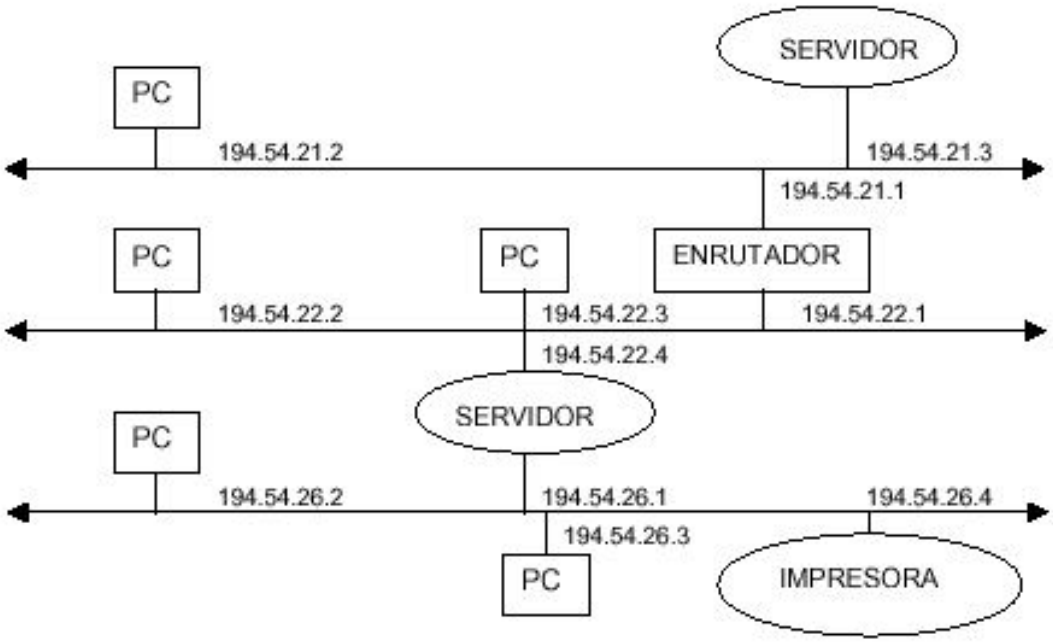

Figura 9. Red mas compleja

Esta es una de las características de la máscara y seguridad de la subred activada en los servidores y enrutadores. La información de una red se enruta a otra red según se requiera.

Como vemos en la gráfica un flujo de datos entre las dos redes inferiores haría prácticamente imposible el enrutamiento entre la inferior y la superior para ello vemos la adición de un enrutador entre estas dos en línea punteada. El acceso entre las dos inferiores se haría por medio del servidor.

Para concluir el diseño de una red debe basarse en directrices y normas. Entre las muchas consideraciones que deben tenerse en cuenta en la planificación de una red, destacan las siguientes:

- ¿Cómo se utilizará la red hoy en día?

- ¿Cómo se utilizará la red el año próximo y el siguiente?

- ¿Qué aplicaciones van a utilizarse en la red?

- ¿Necesitarán los grupos de trabajo dentro de la organización los recursos de red en el futuro?

- ¿Qué tipos y números de estaciones de trabajo habrá en red?

- ¿Cuántos servidores, microcomputadoras y otros sistemas principales habrá en la red?

- ¿Será necesario contar con matrices de discos compartidos y lectores de CD?

- ¿Se centralizará la gestión de la red?

- ¿Estará la red conectada a Internet ó a otras redes corporativas ó constituirá la base de una red de banda ancha?

- ¿Qué otros protocolos utilizará la tecnología de red de trabajo (protocolos IPX, DECNET, LAT, OSI y TCP/IP)?

- ¿Dónde se intercambiarán los datos críticos?. (Determine varias rutas diferentes).

- ¿Cómo crecerá y cambiará la red?

Una vez contestadas todas estas preguntas podrán definir la red. El número de nodos indica los espacios de dirección tipo C necesarios ó en su caso la necesidad de una dirección tipo B.

Tampoco puede olvidarse la posibilidad de conexión a recursos externos. La carga puede distribuirse por segmentos en la red. Intente minimizar el volumen de tráfico que tiene que viajar por la red. Determine la mejor topología de red para satisfacer

los requisitos especificados por el análisis de la red. Para posibilitar el crecimiento de la red el mejor enfoque consiste en determinar la carga máxima y desarrollar una red en la que dicha carga se vea minimizada.

2.2 CONFIGURACION DE LOS SERVICIOS TCP/IP.

La conexión en red TCP/IP de Linux está controlada por un conjunto de archivos de configuración en el directorio /etc. Estos archivos informan a Linux de su dirección IP, nombre del sistema y nombre de dominio, y controlan las interfaces de red. En general el control es realizado por estos tres archivos:

- /etc/host: Asigna los nombres de sistema a las direcciones IP.
- /etc/networks: Asigna los nombres de dominio a la red.
- /etc/rc.d/rc3.d/S10network: Guión que configura y activa sus interfaces Ethernet en el momento del arranque.

2.2.1 Configurando el archivo *letc/host*. Toda computadora en red TCP/IP tiene una dirección IP, un nombre de sistema canónico y opcionalmente, uno ó más

alias de nombre de sistema. El archivo */etc/host* es el método original para asignar nombres de sistema a direcciones IP.

A continuación presentaremos el archivo de una red a Internet tipo B con dos subredes internas tipo C.

```
# /etc/host
#
#
127.0.0.1 localhost
# Esta máquina
166.82.1.21 linux.iutb.com linux # La máquina
local
# Otros sistemas en nuestra red
166.82.1.20 server.iutb.com server #el
servidor
166.82.1.22 estacion.iutb.com #estación de
trabajo.60
166.82.1.10 impre.iutb.com impre #impresora
en la red
166.82.1.1 gateway.iutb.com gateway # El
```

```
enrutador
166.82.1.1 gate-if1 #primera interfaz
166.82.2.1 gate-if2 #segunda interfaz
166.82.1.30 linux2.iutb.com linux2
#portátil vía PLIP
#fin de archivo
```

Observe que la pasarela tiene dos nombres de sistema para la dirección IP. Más es conveniente dar un nombre exclusivo a cada interfaz de red del sistema.
Este archivo es conveniente para redes que no estén conectadas a Internet, es más fácil tener una lista completa de todos los sistemas en /etc/host en lugar de configurar y mantener el DNS.

2.2.2 Configurando el archivo *letc/networks*. Al igual que los sistemas, que pueden tener nombres y direcciones IP, las redes y subredes también pueden denominarse. Esta denominación la maneja el archivo */etc/networks*.

Las direcciones IP en el archivo *networks* sólo incluyen la parte de la dirección de la red más el byte de subred.

A continuación un archivo de ejemplo:

```
#/etc/networks
localnet 127.0.0.0 #software de retorno
iutb-c1 166.82.1 #red de grupo de desarrollo, tipo C
iutb-c2 166.82.2 #red MIS tipo C
#fin de archivo
```

El primero es el nombre de localnet y la dirección IP 127.0.0.0. Si no conecta su sistema Linux a una red TCP/IP ó solo utiliza SLIP ó PPP, sólo tendrá que poner esto en su fichero.

Las líneas siguientes identifican las dos subredes de tipo C que se tienen bajo la red de tipo B, vista en los temas pasados.

2.2.3 Enrutamiento de TCP/IP. El enrutamiento determina la ruta de acceso que toma un paquete en la red hasta alcanzar su destino. Esta ruta se determina comparando la dirección IP, de destino con las tablas de enrutamiento del kernel y transmitiendo el paquete al sistema indicado, que puede ser ó no el destino del paquete. La tabla de enrutamiento de kernel contiene información del tipo “Para ir

a la red X desde el sistema Y, mande el paquete al sistema Z con un coste de 1”, junto con los valores de tiempo de expiración y fiabilidad de dicha ruta.
El primer paso para instalar un enrutamiento en su red es decidir una política de enrutamiento.

En el caso de redes pequeñas y no conectadas a la Internet, bastará con sólo utilizar el comando route para instalar rutas estáticas en cada sistema en el momento del arranque.

Las grandes redes, con numerosas subredes ó redes conectadas a la Internet requieren el uso de un enrutamiento dinámico. El programa de enrutamiento suministra enrutamiento dinámico al comunicarse con programas de enrutado de otros sistemas e instalar rutas basadas en lo que descubre sobre la topología de red.

Una estrategia muy común consiste en combinar el enrutamiento estático con el dinámico. Los sistemas de cada subred utilizan enrutamiento estático para llegar a sus vecinos inmediatos. La ruta predeterminada, la usada por los paquetes que no concuerdan con ninguna otra ruta de la tabla de enrutamiento, esta definida en un

sistema de pasarela que ejecuta enrutamiento dinámico y que posee información sobre el resto del mundo.

El programa /sbin/route maneja la tabla de enrutamiento del kernel y se utiliza para definir rutas estáticas a otras computadoras ó redes, a través de interfaces que *ifconfig* se ha encargado de activar y configurar. Esto se realiza generalmente en el momento de arranque con el guión */etc/rc.d/rc3.d/S10network*. Los parámetros para ese comando puede observarlo en la figura siguiente:

(Ningún)	Provoca la salida de la tabla de enrutamiento local.
-n	Provoca la misma salida anterior pero reemplaza los nombres del sistema por direcciones IP numéricas.
del	Este argumento suprime la ruta de la dirección de destino indicada, de la tabla de enrutamiento.
add	Este argumento añade a la tabla de enrutamiento una ruta a la dirección ó red indicadas.

Figura 10. Parámetros de enrutamiento

Para añadir rutas estáticas a la tabla de encaminamiento, ejecutando el programa de enrutamiento con el argumento add, la sintaxis es:

```
route add [ - net | -host ] addr [gw gateway] [metric cost]
[netmask mask] [dev device]
```

donde `–net` obliga a tratar la dirección como red, `-host` obliga a tratar la dirección como sistema, `addr` es la dirección de destino de la nueva ruta, sea dirección IP ó nombre, `gw` ***`gateway`*** indica que cualquier paquete para esta dirección sea enrutado a través de esta pasarela, `metric` ***`cost`*** establece el campo métrico en la tabla de enrutamiento, `netmask` ***`mask`*** indica la máscara de red de la ruta que se añade (el programa generalmente los especifica automáticamente), `dev` ***`device`*** fuerza la asociación de la ruta con el dispositivo de interfaz de red indicado.

Para suprimir una ruta lo ejecutamos por medio de la sintaxis:

```
[root@lhost ] / # route del [– net –host] addr
```

Los argumentos tienen el mismo significado.

2.3 CONFIGURACION DE UN DNS.

Al principio de la Internet, el número de servidores en la red era reducido. La asignación de nombres y direcciones era una tarea sencilla, pero al pasar el

tiempo el incremento de servidores mantener estos datos en un archivo ASCII resultó siendo inviable.

Para arreglar este problema se creó el sistema de una base de datos distribuida (que abarca toda la red), conocida como BIND, el servidor de dominios de Internet Berkeley. Este sistema también es conocido como servicio de nombres de dominio, sistema de nombres de dominio ó DNS, suministra un nombre de servidor efectivo, relativamente transparente para el proceso de asignación de direcciones IP.

DNS resulta algo difícil de configurar pero luego de esto, el mantenimiento es sencillo.

DNS suministra un mecanismo para la conversión de direcciones IP a nombres mnemónicos que representan sistemas, redes y alias de correos. Para ello, divide todo el espacio de nombres y de IP de Internet en diferentes grupos lógicos. Cada uno de estos grupos tiene autoridad sobre sus propias computadoras y otro tipo de informaciones.

Conceptualmente el DNS puede dividirse en tres partes:

- Espacio del nombre de dominio. Es una especificación de una estructura de un árbol que identifica un conjunto de sistemas y suministra la información, sobre ellos. Conceptualmente, cada nodo en el árbol, tiene una base de datos con información sobre los sistemas que tiene bajo su autoridad. Mediante las pertinentes consultas se extrae información apropiada de esta base de datos. Más sencillamente, es un listado con distintas informaciones, como nombres, direcciones IP, alias de correo, etc., que pueden consultarse en el sistema DNS.

- Servidores de nombre. Son programas que guardan y mantienen los datos localizados en el espacio de nombres de dominio. Cada uno de ellos, posee información completa sobre un subconjunto de espacio de nombres de dominio y retiene información de otras partes. Un servidor de nombres dispone de información completa para su área de autoridad. Esta información autorizada se divide en áreas conocidas como zonas, que pueden dividirse entre distintos servidores de nombres para suministrar el servicio redundante a una zona. Cada servidor de nombres está conectado a otros servidores de nombres que son responsables de zonas distintas. Si entra una petición de información de una zona de la que es responsable un servidor de nombres determinado, el servidor de nombres devuelve la información requerida. Sin embargo, si entra una petición en

busca de información de una zona distinta, el servidor de nombres contacta con el servidor apropiado con autoridad sobre dicha zona.

- Agentes de resolución. Simplemente, son programas o rutinas de biblioteca que extraen información de servidores de nombres en respuesta a una consulta sobre un servidor, en el espacio de nombres de dominio.

2.3.1 Configuración del archivo /etc/host.conf. El primer paso para usar el DNS es configurar la biblioteca del resolvedor en su computador. Las bibliotecas del resolvedor local son configuradas mediante un archivo denominado */etc/host.conf*.

Este archivo informa al resolvedor sobre los servicios que tiene que utilizar y el orden a hacerlo. Este es un simple archivo ASCII que contiene una lista de opciones del resolvedor, una por línea.

Las opciones pueden ser:

- order: Indica el orden en que se prueban los distintos mecanismos de resolución de nombres. Los servicios de resolución indicados se prueban en el orden en que aparecen listados. Se admiten los siguientes mecanismos de

resolución de nombres: *host* (se intenta resolver usando el archivo local */etc/host*), *bind* (consulta un servidor de nombres DNS para resolver el nombre) y *NIS* (utiliza el protocolo de servicio de información de la red para intentar resolver el nombre del sistema).

- alert: Toma *off* u *on* como argumentos. Si se activa on, cualquier intento de falsificación de una dirección IP se registra por medio del servicio *syslog*.

- nospoof: Si se utiliza el resolvedor inverso para comparar un nombre de servidor con una dirección determinada, se resuelve el nombre del servidor que se devuelve para verificar que coincide con la dirección solicitada. Evita la falsificación de las direcciones IP. Se activa al indicar nospoof on. Precaución : El uso de esta opción puede crear una carga insostenible en el servidor.

- trim: Toma un nombre de dominio como argumento. trim elimina el nombre del dominio antes de consultar el nombre en */etc/host* sin especificar el nombre de dominio.

- multi: Toma *off* u *on* como argumento. Se utiliza para determinar si un servidor está autorizado a tener más de una dirección IP indicada en */etc/host*, multi sólo se usa en consultas host. Esta opción no tiene efecto sobre las consultas NIS ó DNS.

A continuación mostraremos un ejemplo de un archivo */etc/host.conf* para el servidor llamado *sistemas.cutb.edu.co*.

```
#Consulta primero nombres vía DNS y después vuelve a
/etc/host
order bind hosts
#no dispone máquinas con múltiple direcciones IP
multi off
#busca falsificaciones de direcciones IP
nospoof on
#y avisar si alguien intenta falsificar
alert on
#Elimina el nombre del dominio para consultas con el servidor
trim sistema.cutb.edu.co
#fin del archivo
```

2.3.2 Configuración del archivo /etc/resolv.conf. Ya configurado el comportamiento básico de la biblioteca del resolvedor, debe configurar alguna información para la parte DNS del mismo. Sólo debe hacerlo si utiliza DNS para resolver nombres del servidor. El archivo /etc/resolv.conf controla la forma en que el resolvedor utiliza DNS para resolver nombres del servidor.

Indica los servidores de nombre DNS que deben contactarse cuando se resuelven un nombre del servidor y el orden en que debe contactarlos. También proporciona el nombre de dominio local y algunas pistas sobre cómo adivinar el nombre de dominio de sistemas que se indican sin él. Las opciones válidas para este archivo son:

- domain: Indica el nombre del dominio local de este sistema. Si no se incluye, el resolvedor tratará de obtenerlo con la llamada del sistema getdomainname().

- nameserver: Indica la dirección IP de un servidor de nombres DNS que debe contactarse para resolver ese nombre. El usuario puede listar hasta tres servidores de nombres, utilizando varias veces la opción nameserver. Los servidores de nombres se prueban en el orden listado. Debería poner en primer

lugar su servidor de nombres más fiable, de forma que no se pierda tiempo consultando un servidor que probablemente estará desconectado.

- search: Da una lista de dominios que deben probarse si no se indica ninguno como parte de un nombre de servidor de consulta. Si no se da ninguna opción de búsqueda, se crea una lista de dominios, utilizando el dominio local y sus superiores correspondientes.

Un ejemplo de este archivo para el servidor *sistemas.cutb.edu.co* es:

```
#/etc/resolv.conf
#especificar el nombre de dominio local
domain sistemas.cutb.edu.co
#especificar el nombre del servidor primario
nameserver xxx.xxx.xxx.xxx
```

Se ha puesto en la dirección IP xxx.xxx.xxx.xxx para señalar una dirección IP de un servidor de nombres el nameserver se debe especificar de esta forma ya que si se coloca el nombre el DNS no sabrá que servidor contactar, señalo que no se colocó ninguna opción de búsqueda, de esta forma si buscamos un servidor de

nombre raven buscará primero raven luego raven.sistemas.cutb.edu.co luego raven.cutb.edu.co luego raven.edu.co y por último raven.co.

2.3.3 Configuración del archivo /etc/named.boot. Hasta ahora hemos visto la instalación de los elementos básicos para la configuración del resolvedor y la forma de indicarle los servidores de nombres que debe contactar.

El servidor de nombres DNS bajo Linux lo proporciona el demonio *named*. Este proceso se arranca normalmente durante el proceso de arranque del sistema y lee la información de configuración en un conjunto de archivos de configuración.

Una vez iniciado el proceso escribe el ID de proceso en */etc/named.pid*. Después se pone a escuchar por el puerto predeterminado que se especifica en */etc/services*.

El archivo */etc/named.boot* es el primer archivo que lee el proceso *named*, este archivo es muy pequeño, pero da las claves para el resto de archivos de configuración y otros servidores de nombres. En este archivo los comentarios se marcan con punto y coma (;) y continúan hasta el final de la línea.

He aquí varias opciones que pueden listarse en el archivo *named.boot*:

- directory: Es el directorio donde se encuentran los archivos de zona DNS. Puede indicar distintos directorios utilizando repetidamente la opción directory. También puede asignar nombres de rutas de acceso de los archivos asociados a estos directorios.

- primary: Toma un nombre de dominio y uno de archivo como argumentos. La opción primary autoriza a named en el dominio indicado y hace que éste cargue la información de zona del archivo indicado.

- secondary: Esta opción indica a named que actúe como servidor secundario en el dominio indicado. Toma como argumentos un nombre de dominio, una lista de direcciones y un nombre de archivo. named intenta transferir la información de zona de los servidores indicados en la lista de direcciones y almacena esta información en el archivo indicado en la línea de opciones. Si named no puede contactar con ninguno de los servidores, intenta recuperar la información del archivo de zona secundario.

- cache: Instala información de caché de named. Toma como argumentos un nombre de dominio y uno de archivo. El nombre de dominio se indica.

normalmente mediante un punto (.). El archivo contiene un conjunto de registros, conocidos como indicios del servidor, que listan información sobre los servidores en esta lista si no puede resolver una dirección desde su información local.

- forwarders: Toma una lista de servidores de nombres como argumentos. Indica al servidor de nombres local que trate de contactar con los servidores en esta lista si no puede resolver una dirección desde su información local.

- slave: Convierte al servidor de nombres local en un servidor esclavo. Si se introduce la opción slave, el servidor local intenta resolver los nombres DNS mediante consultas recursivas. Simplemente remite la petición a uno de los servidores listados en la línea de opciones *forwarders*.

Existen opciones adicionales para este archivo consulte man named en su shell para más ayuda.

A continuación un listado con un archivo de ejemplo del archivo *named.boot*:

```
; archivo named.boot
;ejemplo de este archivo para sistemas.cutb.edu.co
;
```

```
directory /var/named
;Los archivos de trabajo en /var/named
;
cache . named.ca
;instala la información de cahé y cargue la información del
;servidor raíz desde el archivo named.ca
;
primary sistemas.cutb.edu.co named.hosts
;indica que este servidor tiene propiedad primaria en el
;dominio sistemas.cutb.edu.co. Los registros de zona y
;sistema se encuentra en named.hosts
;
primary 197.198.199.in-addr.arpa named.rev
;indica que también tiene prioridad en la zona primaria
;197.198.199.in-addr.arpa, con información de zona en el
;archivo named.rev. La sintaxis es la forma de cómo compara
;direcciones IP con DNS, por la forma que fue desarrollada
;esta comparación se hace necesaria una línea para llevar a
;cabo la resolución inversa.
```

2.3.4 Archivos de bases de datos y registro de recursos. Toda la información acerca de los archivos de la base de datos de named se almacena en un formato conocido como registro de recursos. Cada registro de recursos posee un tipo asociado a él, que informa de la función del registro.

Un registro de recursos es la unidad de información más pequeña que maneja named. La sintaxis de estos registros es compleja, y la dispersión de los archivos hace que la configuración de estos archivos maestros sea confusa, por ende se convierte en problema principal para servidores DNS. Los registros de recursos utilizan una sintaxis general, formada por todos los tipos posibles de registros.

Sin embargo, existen varias partes del registro que son opcionales, dependiendo del tipo de registro, y pueden tomar un valor predeterminado si no se indican. El formato básico del registro de recurso es `[owner] [ttl] [class] type data`, estos argumentos son explicados en la tabla siguiente:

Campo	Descripción
owner	El nombre de dominio ó servidor al que se aplica el registro. Si no se especifica ningún nombre, se supone el nombre del dominio del registro del recurso anterior.

ttl	El campo de tiempo de expiración (en segundos). Este campo informa sobre el tiempo en que tendrá validez la información del registro de conexión de haberse recuperado desde el servidor DNS. Si no se da el valor de conexión a *ttl*, se utiliza el *ttl* mínimo del último registro de comienzo de autoridad.
class	Indica una clase de dirección de conexión en red. Para las redes TCP/IP, use el valor *IN*. Si no se asigna la clase, se usa la del registro de recurso anterior.
type	Lista el tipo de registro de recurso. Este valor es obligatorio.
data	Indica los datos asociados al registro de recursos. Este valor es obligatorio. El formato del campo *data* depende de lo que contenga el campo *type*.

Tabla 3. Campos del registro de recursos

Además de estos campos existen otros adicionales que podrá encontrar ejecutando `man named`, ó buscando en los RFC asociados.

Sabiendo ya los formatos pasemos a los tipos de registros de recursos más usados. Podemos verlos en la siguiente tabla.

Tipo	Descripción
A	Es un registro de dirección. Asocia un nombre de sistema a una dirección. El campo de datos la guarda la dirección en un formato decimal con puntos. Sólo puede haber un registro *A* para un determinado servidor, ya que este registro se considera información autorizada. Cualquier asignación de nombre de servidor o dirección adicional para este servidor, debe darse utilizando el tipo *CNAME*.
CNAME	Este campo asigna un alias a un servidor con su nombre canónico, es decir el nombre indicado en el registro *A* de este sistema.
HINFO	Proporciona información sobre un sistema. El campo de datos guarda la información de hardware y software de un determinado servidor. En este caso, se trata simplemente de una cadena de texto en formato libre, por lo que puede incluir cualquier información que su hardware pueda conocer.
MX	Instala un registro de intercambio de correo. El campo de datos

	guarda un valor entero de preferencia, seguido del nombre del servidor. Los registros *MX* indican a un transporte de correo que envíe correo a otro servidor que sabe cómo hacer llegar dicho correo a su final.
NS	Apunta a un servidor de nombres de otra zona. El campo de datos del registro de recursos *NS* contiene el nombre del DNS y el del servidor de nombres. El usuario debe indicar un registro *A* y comparar el nombre del servidor con la dirección del servidor de nombres.
PTR	Asigna direcciones a nombres, como el dominio *in-addr.arpa*. El nombre del servidor debe ser el canónico.
SOA	Informa al servidor de nombres que todos los registros de recursos que le siguen están autorizados en dicho dominio (SOA es la abreviatura de comienzo de autoridad). El campo de datos está puesto entre paréntesis () y normalmente se trata de un campo multilineal. El campo de datos del registro SOA contiene las siguientes entradas:

	➢ origin: El nombre canónico del servidor de nombres primario de este dominio. Usualmente, se asigna como un nombre de dominio absoluto que termina por un punto (.), para que el proceso demonio named no lo modifique. ➢ contact: El contacto por correo electrónico de la persona responsable del mantenimiento de este dominio. Puesto que el carácter arroba (@) tiene un significado especial en registros de recursos, se substituye por un carácter de punto (.). Si la persona responsable de mantener la información de zona sobre *sistemas.cutb.edu.co* es José, la dirección de contacto será *jose.sistemas.cutb.edu.co*. ➢ serial: El número de versión del archivo de información de zona, que se refiere en enteros. Lo utilizan servidores de nombres secundarios para determinar cuándo se ha modificado el archivo de registro de zona. El usuario debería incrementar este valor en 1 (uno), siempre que modifique el archivo de información.

	➢ refresh: La cantidad de tiempo expresada en segundos que debe esperar un servidor secundario antes de comprobar el registro *SOA* del servidor de nombres primario. Los registros *SOA* no cambian con frecuencia, por lo que normalmente puede ajustarse este valor para que corresponda sea más ó menos a 1 (un) día. ➢ retry: Este es el tiempo expresado en segundos que espera un servidor secundario para volver a intentar una información al servidor primario, si esté no está disponible. Por lo general, debería fijarse en torno a un par de minutos. ➢ expire: Este es el tiempo expresado en segundos que espera un servidor secundario antes de desechar la información de zona si no ha podido contactar con el servidor primario. Este número debe ser grande, del orden de unos 30 (treinta) días. ➢ minimum: Este es el valor *ttl* predeterminado para registros de recursos que no incluyan un *ttl*. Si su red no cambia mucho, puede especificarse un valor grande, como un par de semanas. Siempre puede sobrescribirlo, indicando un valor *ttl* en sus registros de

	recursos.

Tabla 4. Tipos en el registro de recursos

2.3.5 Configuración del archivo *`/etc/named.hosts`*. En el archivo *named.boot* colocamos named.hosts como el archivo que contiene información sobre su dominio local, sistemas.cutb.edu.co. Podría haber designado otro archivo con solo designarlo en la línea primary de named.hosts. Este archivo contiene la información de autorizaciones sobre los servidores en la zona de autoridad, sistemas.cutb.edu.co.

A continuación mostraremos un listado de ejemplo del archivo /etc/named.hosts con varios registros de recursos:

```
;archivo named.hosts
;
@ IN SOA ns.sistemas.cutb.edu.co
jose.sistemas.cutb.edu.co.
;de comienzo especificamos el nombre canónico del servidor de
;nombres primarios. Luego colocamos el mail del contacto
;sustituyendo la @ (arroba) por un punto ya que en los
```

```
;registros de recursos este carácter es usado con otro fin.
(
6 ; número serial
86400 ; refresh cada 24 horas
300 ; retry cada 5 minutos
2592000 ; expire cada 30 dias
86400 ; minimum 24 horas
)
IN NS ns.sistemas.cutb.edu.co.
;
;su propio dominio
;
@ IN A 199.198.197.1.70
;permite a otros usuarios ingresar al servidor por el nombre
;canónico refriéndose al IP.
IN MX 100 mail.sistemas.cutb.edu.co
IN HINFO PC-486 Linux
;
;su nameserver primario
;
ns IN A 199.198.197.1
```

```
nameserver IN CNAME ns.sistemas.cutb.edu.co

;

;otros servidores

;

mail IN A 199.198.197.2

catedra IN A 199.198.197.3

IN MX 100 mail.sistemas.cutb.edu.co

;configura los registros de dirección de otros servidores al

;final especifica el envío de correos al servidor con

;el nombre de usuario especificado.

;

;el servidor local

;

localhost IN A 127.0.0.1

;fin de archivo
```

2.3.6 Configuración del archivo *named.rev*. El archivo *named.rev* es muy similar a *named.hosts*, salvo que trabaja esencialmente al revés. Asigna direcciones a nombres de sistema.

A continuación un ejemplo de configuración de este archivo:

```
;archivo named.rev
;
```

```
;
@ IN SOA ns.sistemas.cutb.edu.co.
jose.sistemas.cutb.edu.co.
(
6 ;número serial
86400 ;refresh cada 24 horas
300 ;retry cada 5 minutos
2592000 ;expire cada 30 días
86400 ;minimum 24 horas.
)
;.71
;
;indica el nombre de dominio de servidor de nombres
IN NS ns.sistemas.cutb.edu.co
;
;
;asigna al réves direcciones IP de los servidores en su
;dominio
;registro de resolución inversa de servidores, aquí menciona
;que el servidor 199.198.197.2 es usado por mail, el IP
;199.198.197.4 es usado por profesores así sucesivamente.
;
1 IN PTR ns.sistemas.cutb.edu.co
2 IN PTR mail.sistemas.cutb.edu.co
3 IN PTR catedra.sistemas.cutb.edu.co
4 IN PTR profesores.sistemas.cutb.edu.co
;
;fin de archivo
```

2.3.7 Configuración del archivo *`/etc/named.ca`*. Tal como habiamos dicho la operación de caché es muy importante. Afortunadamente el archivo *named.ca* que configura el caché, es también uno de los archivos named más sencillos. Tan solo lista los servidores de nombres raíz de los distintos dominios con sus respectivas direcciones IP.

Contiene un par de indicadores de campo especiales que informan a named que se trata de servidores raíz.

Probablemente, bastará con que copie el formato del archivo *named.ca* que se muestra en el siguiente ejemplo. Para obtener una lista actual y completa de los servicios de nombres raíz, use la utilidad nslookup.

A continuación el archivo de ejemplo:

```
;archivo named.ca
;
. 99999999 IN NS NS.NIC.DDN.MIL.
99999999 IN NS NS.NASA.GOV.
99999999 IN NS KAVA.NSC.SRI.COM
99999999 IN NS TERP.UMD.EDU.
99999999 IN NS C.NYSER.NET.
99999999 IN NS NS.INTERNIC.NET.
;
NS.NIC.DDN.MIL. 99999999 IN A 192.112.36.4
NS.NASA.GOV. 99999999 IN A 128.102.16.10
```

```
KAVA.NSC.SRI.COM. 99999999 IN A 192.33.33.24
TERP.UMD.EDU. 99999999 IN A 128.8.10.90.72
C.NYSER.NET. 99999999 IN A 192.33.4.12
NS.INTERNIC.NET. 99999999 IN A 198.41.0.4
```

Como habrá podido comprobar este archivo asigna simplemente los registros de asignación de nombres a las direcciones apropiadas.

2.3.8 Resolución de problemas. El sistema DNS es muy complejo. Existen muchas cosas que el usuario puede ejecutar erróneamente y que pueden provocar un funcionamiento inadecuado del servidor. Aunque los problemas parezcan idénticos puede observarse que cada problema puede tener una solución distinta aunque la gran mayoría son errores de síntaxis. Para evitar esto usted puede seguir los siguientes consejos:

- Asegúrese que indica correctamente los nombres del sistema en sus archivos de configuración DNS. Si es un nombre de sistema absoluto, asegúrese de que finaliza con un punto (.).

- Tenga mucho cuidado con los nombres utilizados por los registros *SOA* y *CNAME*. Si comete errores aquí, estos registros pueden redirigir consultas de nombres de servidor a computadoras que no existen.

- Asegúrese de aumentar el número serial cada vez que realice cambios en sus archivos de configuración. Si olvida esto el DNS no podrá volver a leer el archivo.

- Asegúrese que introduce la dirección IP correcta de los registros A y compruebe que coincide con el archivo */etc/hosts*, si tiene uno. Compruebe también que el nombre de DNS y la dirección IP coinciden con la información de resolución inversa correspondiente de *named.rev*.

- Su mejor herramienta para detectar errores es el comando *nslookup*. Use esta herramienta para probar a fondo su servidor DNS. Igualmente, realice resoluciones directas e inversas con cada una de las direcciones de su base de datos DNS para estar seguro de que todos los nombres y direcciones están correctos.

2.4 EL SUPER SERVIDOR INETD

Frecuentemente, los servicios son llevados a cabo por los llamados demonios. Un demonio es un programa que abre un determinado puerto, y espera a recibir peticiones de conexión. Si se recibe una petición de conexión, lanza un proceso hijo que aceptará la conexión, mientras el padre continúa escuchando a la espera de más peticiones. Este concepto tiene el inconveniente de que por cada servicio ofrecido, se necesita ejecutar un demonio que escuche por su puerto a que ocurra una conexión, lo que generalmente significa un desperdicio de recursos de sistema como, por ejemplo, de espacio de intercambio.

Por ello, casi todas las instalaciones corren un “super-servidor'' que crea sockets para varios servicios, y escucha en todos ellos simultáneamente usando la llamada al sistema select. Cuando un nodo remoto requiere uno de los servicios, el super-servidor lo recibe y llama al servidor especificado para ese puerto.

El super-servidor usado comúnmente es inetd, el demonio Internet. Es iniciado en tiempo de arranque del sistema, y toma la lista de servicios que debe tratar de un fichero de arranque denominado */etc/inetd.conf*. Aparte de esos servidores invocados por inetd, hay varios servicios triviales que el propio inetd se encarga de

llevar a cabo denominados servicios internos. Entre ellos, el *chargen* que simplemente genera una cadena de caracteres, y el daytime que devuelve la idea del sistema de la hora del día.

2.4.1 Configuración del super-servidor inetd. Para la configuración de los servicios proporcionados por el inetd se hace necesario configurar el archivo */etc/inetd.conf*, punto que describiremos a continuación. El archivo *inetd* consta de líneas con el siguiente formato:

- Servicio tipo/socket protocolo espera usuario servidor linea_de_comando, donde:
  - Servicio: Proporciona el nombre del servicio. El nombre del servicio debe ser traducido a un número de puerto consultando en el fichero */etc/services*.
  - Tipo/socket: Especifica un tipo de socket, ya sea *stream* (para protocolos orientados a la conexión) o *dgram* (para protocolos no orientados a la conexión). Servicios basados en TCP deberán, por lo tanto, usar siempre *stream*, mientras que los servicios basados en UDP deberán usar siempre *dgram*.

- Protocolo: Indica el protocolo de transporte usado por el servicio. Este debe ser un nombre de protocolo válido que se pueda encontrar en el fichero *protocols*.

- Espera: Esta opción se aplica sólo en sockets de tipo *dgram*. Puede tomar los valores *wait* o *nowait*. Si se especifica *wait*, *inetd* ejecutará sólo un servidor cada vez para el puerto especificado. De otro modo, continuará escuchando por el puerto inmediatamente después de ejecutar el servidor.

  Esto es útil para servidores ``single-threaded'' que leen todos los datagramas que entran hasta que no llegan más, y después acaban. La mayor parte de los servidores RPC son de éste tipo y deberán por ello especificar *wait*. El otro tipo de servidores, los ``multi-threaded'', permiten un número ilimitado de instancias corriendo concurrentemente; éstos son raramente utilizados. Estos servidores deberán especificar *nowait*. Con sockets de tipo *stream* se deberá especificar siempre *nowait*.

- Usuario: Este es el identificador de *login* del usuario bajo el que se ejecutará el proceso. Por lo general, éste suele ser el usuario *root*, aunque algunos servicios pueden usar diferentes. Es una buena idea aplicar aquí el

principio del menor privilegio, que indica que uno no debe ejecutar un comando bajo una cuenta privilegiada si el programa no lo requiere para funcionar correctamente.

Por ejemplo, el servidor de noticias *NNTP* se ejecutará como *news*, mientras que otros servicios que podrían significar un riesgo para la seguridad (como *tftp* o *finger*) son normalmente ejecutados como *nobody*.

- Servidor: Proporciona el camino completo del programa servidor a ejecutar. Los servicios internos son marcados con la palabra internal.

- Línea_de_comandos: Esta es la línea de comando a pasar al servidor. Esto incluye el argumento 0, es decir, el nombre del comando. Normalmente, éste será el nombre de programa del servidor, salvo que el programa se comporte de forma distinta cuando se le invoque con un nombre diferente. Este campo se deja vacío para los servicios internos.

Podemos ver un ejemplo de un archivo *inetd.conf*:

```
#
#archivo inetd.conf
#servicios del inetd
```

```
#
ftp stream tcp nowait root /usr/sbin/ftpd in.ftpd –l
telnet stream tcp nowait root /usr/sbin/telnetd in.telnetd -b
etc/issue
#finger stream tcp nowait bin /usr/sbin/fingerd in.fingerd
#tftp dgram udp wait nobody /usr/sbin/tftpd in.tftpd
#tftp dgram udp wait nobody /usr/sbin/tftpd in.tftpd
/boot/diskless
#login stream tcp nowait root /usr/sbin/rlogind in.rlogind
shell stream tcp nowait root /usr/sbin/rshd in.rshd
exec stream tcp nowait root /usr/sbin/rexecd in.rexecd
#
#servicios internos
#
daytime stream tcp nowait root internal
daytime dgram udp nowait root internal
time stream tcp nowait root internal
time dgram udp nowait root internal
echo stream tcp nowait root internal
echo dgram udp nowait root internal
discard stream tcp nowait root internal
```

```
discard dgram udp nowait root internal
chargen stream tcp nowait root internal
chargen dgram udp nowait root internal
#
#fin del archivo
```

El archivo configura para el acceso a ciertos servicios y a otros no, las líneas antepuestas por el carácter # son comentarios al anteponer este carácter a una línea de un servicio deshabilitamos el servicio de nuestro sistema.

2.5 LA HERRAMIENTA DE CONTROL DE ACCESO *tcpd*.

Como abrir un ordenador al acceso en red implica muchos riesgos de seguridad, las aplicaciones están diseñadas para protegerse ante varios tipos de ataques.

Algunas de estas aplicaciones, sin embargo, pueden ser reventadas (lo que quedó claramente demostrado con el *RTM Internet worm*), o pueden no distinguir entre un nodo seguro cuyas peticiones de un servicio particular deberían ser aceptadas, y otro nodo que no lo es y cuyas peticiones deberían ser rechazadas.

Así, uno podría querer limitar el acceso a esos servicios solamente a los “nodos de confianza", lo cual es imposible con la configuración usual, donde *inetd* proporciona un servicio a todos los clientes, o a ninguno.

Una herramienta útil para esto es *tcpd*, el denominado demonio envoltorio. Para los servicios TCP que quiera monitorizar o proteger, éste es invocado en lugar del programa servidor. *tcpd* informa de la petición al demonio *syslog*, chequea si el nodo remoto está autorizado para usar ese servicio, y sólo si la respuesta es satisfactoria, ejecutará el programa servidor real. Observe que esto no funciona con servicios basados en *UDP*.

El control de acceso está implementado mediante dos ficheros llamados */etc/hosts.allow* y */etc/hosts.deny*. Estos ficheros contienen entradas permitiendo y denegando acceso, respectivamente, para ciertos servicios y nodos.

Cuando *tcpd* trata una petición de un servicio como finger de un nodo cliente denominado *universidad.colombia.com*, busca en *hosts.allow* y *hosts.deny* (en éste orden) una entrada en la que el servicio y el nodo cliente coincidan. Si la entrada coincidente aparece en *hosts.allow*, se garantiza el acceso, sin importar lo que haya en *hosts.deny*. Si la coincidencia se encuentra en *hosts.deny*, la petición

se rechaza cerrando la conexión. Si no hay coincidencia en ninguno, la petición es aceptada.

Las entradas en los ficheros de acceso tienen la siguiente estructura:

```
lista_servcios : lista_nodos [:cmd_shell]
```

Donde `lista_servicios` es una lista de nombres de servicios de */etc/services*, ó la palabra clave `ALL`. Para especificar todos los servicios excepto *finger* y *tftp*, usa “ALL EXCEPT finger, tftp". `lista_nodos` es una lista de nombres de nodos o direcciones IP, o las palabras clave `ALL`, `LOCAL`, o `UNKNOWN`.
`ALL` hace coincidir todos los nodos mientras que `LOCAL` hace coincidir todos los nombres de nodos que no contengan un punto, `UNKNOWN` hace coincidir todos los nodos cuya búsqueda de nombre ó dirección falló.

Un nombre que comienza por un punto incluye a todos los nodos cuyo dominio es el mismo a ese nombre. Por ejemplo: `.colombia.com` coincidirá con `universidad.colombia.com`.

También hay formas de especificar direcciones de red IP y números de subred.

Para denegar acceso a los servicios *finger* y *tftp* a todos los nodos menos a los locales, ponga lo siguiente en */etc/hosts.deny*, y deje */etc/hosts.allow* vacío:

```
in.tftpd, in.fingerd : ALL EXCEPT LOCAL, .su.dominio
```

Quedaría en el archivo de la siguiente manera:

```
in.tftpd: ALL EXCEPT LOCAL, .colombia.com:\
echo “petición de %d@%h” >> /var/log/finger.log;\
if [%h != “universidad.colombia.com”]; then \
finger –l @%h >> /var/log/finger.log \
fi
```

El campo opcional `cmd_shell` puede contener un comando de shell para que sea invocado cuando una búsqueda coincida con la entrada. Esto es útil para establecer trampas que puedan delatar a atacantes potenciales. Los argumentos `%h` y `%d` son expandidos por tcpd al nombre del nodo cliente y al nombre del servicio, respectivamente.

2.6 CONFIGURACION DE LOS ARCHIVOS /etc/services y /etc/protocols.

Los números de puerto en los que se ofrecen ciertos servicios “estándar" están definidos en el RFC “Números Asignados". Para permitir a los programas cliente y servidor convertir nombres de servicios en estos números, al menos una parte de la lista es mantenida en cada nodo; está almacenada en un fichero llamado */etc/services*. Una entrada se construye así:

```
servicio puerto/protocolo [aliases]
```

Donde servicio especifica el nombre del servicio, puerto define el puerto por el que se ofrece el servicio, y protocolo define qué protocolo de transporte se usa.

Comúnmente, éste es udp o tcp. Es posible que un servicio sea ofrecido a más de un protocolo, lo mismo que es posible ofrecer distintos servicios por el mismo número de puerto, siempre que el protocolo sea distinto. El campo aliases permite especificar nombres alternativos para el mismo servicio.

El archivo */etc/services* puede quedar:

```
#archivo /etc/services
```

```
#
#servicios conocidos
echo 7/tcp
echo 7/udp
discard 9/tcp sink null
discard 9/udp sink null
daytime 13/tcp
daytime 13/udp
chargen 19/tcp ttytst source
chargen 19/udp ttytst source
ftp-data 20/tcp
ftp 21/tcp
telnet 23/tcp
smtp 25/tcp.77
nntp 119/tcp readnews
#
#servicios UNIX
#
exec 512/tcp
biff 512/udp comsat
login 513/tcp
```

```
who 513/udp whod
shell 514/tcp cmd
syslog 514/udp
printer 515/tcp spooler
router 520/udp router routed
#fin de archivo
```

Observe que, por ejemplo, el servicio echo es ofrecido en el puerto 7 tanto para *TCP* como para *UDP*, y que el puerto 512 es usado para dos servicios diferentes, el demonio *COMSAT* (que notifica a los usuarios de correo recién llegado, vea *xbiff(1x)*), mediante *UDP*, y la ejecución remota (*rexec(1)*), usando *TCP*.

Similar al fichero de servicios, la librería de red necesita una forma de convertir nombres de protocolo (por ejemplo, los usados en el fichero *services*) a números de protocolo entendibles por el nivel IP en otros nodos. Esto se hace buscando el nombre en el fichero */etc/protocols*.

Contiene una entrada por línea, cada una conteniendo un nombre de protocolo y el número asociado. Necesitar modificar éste fichero es todavía más improbable que tener que hurgar en */etc/services*.

A continuación un ejemplo del archivo /etc/protocols:

```
#
#
#archivo /etc/protocols
#
ip 0 IP
icmp 1 ICMP
igmp 2 IGMP
tcp 6 TCP
udp 17 UDP
raw 255 RAW
#
#fin de archivo
```

2.7 EL PROTOCOLO DHCP.

El DHCP fue desarrollado por el IETF como sucesor de otro protocolo denominado BOOTP.

Las mejoras del DHCP con respecto al BOOTP básicamente se pueden clasificar en dos grupos: por un lado, permite que un ordenador obtenga toda la información que necesita en un único mensaje, (dirección IP, máscara de subred, etc.) con el consiguiente ahorro de tráfico y tiempo; y por otro, posibilita que una máquina disponga de una dirección IP de una rápida y dinámica. Inicialmente el protocolo RARP (Protocolo inverso de asociación de direcciones) era el único mecanismo existente para que una máquina obtuviera una dirección IP.

Los motivos por los que se hacía necesario que una máquina necesitar obtener su dirección IP de otra eran diferentes a los actuales; en el momento en el que se desarrolló RARP se hizo para posibilitar que los ordenadores sin disco duro. Debido entre otras cosas, el nivel tan bajo en el que trabajaba RARP y a la poca información que contenía un mensaje RARP, pronto se vio la necesidad de realizar modificaciones en este protocolo desarrollándose, de esta forma, el protocolo BOOTP y, seguidamente, el protocolo DHCP.

Las diferencias entre estos dos últimos protocolos son mínimas, por lo que su parecido es elevado. El protocolo BOOTP se basa en UDP e IP para funcionar, por lo que debe ser tratado como un protocolo de nivel aplicación.

El funcionamiento es el siguiente:

Cuando una máquina necesita su dirección IP utiliza el protocolo UDP para transportar la petición de dirección; cada mensaje UDP es encapsulado en un datagrama IP. Aunque parece paradójico, se puede decir que se utiliza IP para determinar una dirección IP. Probablemente se esté preguntando, y sobre todo si tiene en cuenta el formato de un datagrama IP, cómo una máquina sin dirección IP puede rellenar campos como el dirección IP de destino.

Pues bien, el campo de dirección IP del destino se rellena con una dirección de multidifusión:

255.255.255.255. Cuando un datagrama llega a un servidor de direcciones IP con esta dirección IP de destino, el servidor sabe que es para él (la dirección dice que es para todas las máquinas) y realiza las operaciones pertinentes.

Pero no se acaba aquí. Ya se sabe cómo hacer llegar una petición de dirección IP a un servidor de direcciones IP, pero ¿cómo hacer llegar la dirección IP el servidor a la máquina cliente, si ésta todavía no tiene dirección?. Pues bien, lo que se hace es poner como dirección destino la misma dirección de multidifusión.

De la misma forma de RARP se quedó pequeño en poco tiempo, el protocolo BOOTP, debido a alguna de sus características y a la evolución de las necesidades, también se ha quedado pequeño en determinadas circunstancias, por lo que surgió la necesidad de crear un nuevo protocolo: DHCP.

BOOTP fue desarrollado para un entorno relativamente estático en el que cada máquina tenía configuración permanente; teniendo en cuenta esto, el administrador creaba un fichero de configuración BOOTP para cada máquina, que sufriría, debido a los pocos cambios que se hacen en la red, pocas modificaciones, por lo que su tarea no se ve incrementada por esto. Pero las cosas han ido cambiando poco a poco, y las máquinas en la actualidad sufren frecuentes modificaciones, no sólo de ubicación sino de carácter interno.

Como el protocolo BOOTP no se adapta a estos cambios, surgió la necesidad de crear un nuevo protocolo más flexible y dinámico; así nació DHCP.

Aunque hasta el momento únicamente se ha hablado de la instalación de DHCP como servidor de direcciones IP, en realidad, cuando una máquina se conecta,

busca algo más que una dirección IP: busca otros datos que le permiten configurarse para un correcto funcionamiento en red.

2.7.1 Funcionamiento del DHCP. Para disponer de asignación dinámica de direcciones IP en una red, se deben configurar tanto los clientes como un servidor de direcciones. Al servidor de direcciones se le asigna un rango de direcciones para otorgar cada vez que realice una petición; la asignación, que posteriormente el servido haga de estas direcciones, puede ser manual ó automática.

Se tratará de una asignación manual cuando el administrador asocie directamente a una máquina concreta una dirección IP, de forma que, cuando esta máquina solicite una IP, siempre se le dará la misma: la establecida por el administrador.

Esta es la forma que tiene que comportarse el protocolo BOOTP. Se dice que la asignación es automática como hace DHCP, o lo que es lo mismo dinámica, cuando es el servidor de direcciones IP el qué decide que dirección dar a una máquina cuando se conecta. Este tipo de asignación es temporal, ya que la asignación no es para siempre sino únicamente mientras que la máquina esté conectada. La próxima vez que se conecte recibirá una nueva IP, que puede o no

coincidir con la anterior. El tiempo de asignación de la dirección IP puede ser centrada de acuerdo a la función que desempeña la máquina y el entorno de la red.

2.7.2 Características del DHCP. A continuación se muestran las principales características de DHCP, que han hecho que sea un protocolo muy extendido, implementado en los principales sistemas operativos, routers y periféricos de red.

Estas características serán detalladas en los siguientes apartados: Se asienta sobre el protocolo UDP (puerto 67 el servidor y 68 el cliente). Utilizando Broadcast (dirección IP de difusión límite 255.255.255.255), para la comunicación con el cliente. La responsabilidad de la confiabilidad de la comunicación recae sobre el cliente (mediante time-out y retransmisión en caso de error). Diseñado para ser compatible con BOOTP.

Utiliza el mismo formato de mensaje. Con arquitectura cliente-servidor, permitiendo el uso de agentes BOOTP relé si el servidor y el cliente están en una red diferente.

El servidor admite tres tipos de configuraciones de direcciones IP:

- Estática: Se configura en el servidor la dirección de red que se corresponde con la dirección LAN del cliente (equivalente a BOOTP).

- Dinámica, por tiempo ilimitado. Se indica un rango de direcciones que se asignan a cada cliente de carácter permanente, hasta que el cliente la libera.

- Dinámica, arrendada. Las direcciones se otorgan por un tiempo limitado. Un cliente debe renovar su dirección para poder seguir utilizándola.

Cada vez que arranca un cliente, debe volver a solicitar una dirección. El servidor debe recordar si el cliente ya tenía una dirección para reasignársela. Si el servidor no responde, el cliente podrá usar la dirección, mientras no caduque su tiempo. La asignación de direcciones IP se configurará un modo u otro, dependiendo de cada situación.

Puede interesar un direccionamiento estático para clientes sin disco o por facilidades administrativas, pero controlando la asignación de cada dirección a cada cliente (es más cómodo para el administrador configurar un servidor, que

cada cliente; interesa el direccionamiento estático para evitar que se conecten clientes no identificados o por otras razones, como la configuración de DNS).

El direccionamiento dinámico por tiempo ilimitado se utiliza cuando el número de clientes no varía demasiado, facilitando mucho la tarea del administrador. El arrendamiento de direcciones se emplea para racionar las direcciones IP, minimizando el coste administrativo.

En función de la frecuencia de inserciones/eliminaciones de clientes y de la cantidad de direcciones disponibles se concederá un mayor o menor tiempo de arrendamiento. El tiempo será bajo (ej. 15 minutos) si se conectan/desconectan los clientes con mucha frecuencia e interesa que esté disponible el máximo número de direcciones. Por el contrario se utilizará un tiempo largo para que cada cliente mantenga su dirección IP (ej. En una Universidad un tiempo de 4 meses, tiempo máximo que está desconectado en vacaciones para asumir que el cliente ya no está en la red). Un portátil puede tener una dirección permanente o de larga duración en su red habitual de trabajo y tiempos cortos en otras redes.

2.7.3 Tipos de mensajes DHCP. DHCP no sólo sirve para asignarle una dirección IP a una máquina, sino que entre muchas otras funciones pueden dar toda la

configuración TCP/IP. Para poder llevarla a cabo incorpora diferentes tipos de mensajes, en la tabla siguiente podemos observar 7 tipos de mensajes que son usados por el cliente y el servidor DHCP para el intercambio de información.

Como vemos en la tabla existen paquetes para todas las situaciones que pueden darse durante la conversación de un cliente DHCP con un servidor.

Tipo de mensaje	**Descripción**
Dhcpdiscover	La primera vez que trata de iniciarse un cliente *DHCP* en la red, solicita información de la dirección IP a un servidor *DHCP* mediante la difusión de un paquete dhcpdiscover, la dirección IP de origen es 0.0.0.0 porque el cliente todavía no tiene ninguna dirección.
Dhcpoffer	Cuando el servidor *DHCP* recibe la petición, selecciona una dirección IP no asignada del rango de direcciones disponibles y se la ofrece al cliente *DHCP*. En la mayoría de las ocasiones, el servidor *DHCP* también devuelve información de configuración de TCP/IP adicional, como la máscara de subred y la puerta de enlace predeterminada, en este paquete.

	Más de un servidor *DHCP* puede responder por lo que el cliente acepta el primero que recibe.
Dhcprequest	Cuando un cliente recibe un paquete *dhcpoffer*, responde enviando por difusión un paquete *dhcprequest* que contiene la dirección ofrecida.
Dhcpdecline	Este es un mensaje que envía el cliente *DHCP* al servidor, indicándole que los parámetros de la configuración ofrecida no son válidos.
Dhcpack	El servidor *DHCP* reconoce el paquete *dhcprequest* del cliente enviando un *dhcpack*.
Dhcprelease	Es un mensaje del cliente *DHCP* al servidor que libera la dirección IP y cancela cualquier concesión que se mantenga.
Dhcpnack	Si el cliente no puede utilizar la dirección IP porque ya no es válida o la esta utilizando otra máquina, el servidor *DHCP* responderá con un paquete de este tipo.

Tabla 5. Tipos de mensajes del DHCP

A modo de resumen se podría decir que durante el inicio del sistema, cuando éste necesita información de configuración TCP/IP, la conversación entre cliente y servidor DHCP sigue el siguiente esquema:

- El cliente DHCP solicita una dirección IP al servidor.
- El servidor oferta una dirección IP al cliente.
- El cliente responde aceptando la dirección IP.
- El servidor confirma la concesión de la dirección.

2.7.4 Relación entre el protocolo DHCP y los nombres de dominio (DNS). Se ha visto que DHCP es capaz de asignar direcciones IP y dar parámetros de configuración a las máquinas que así lo solicitan; sin embargo, en ningún momento se ha hablado del nombre de la máquina y de cómo se resuelve esta correspondencia.

Ya sabe que una máquina tiene una dirección IP y un nombre asociado, de tal forma que se puede acceder a ella a través del nombre o bien por medio de la dirección IP, y que aunque internamente se trabaja con las direcciones de las máquinas, al nivel de usuario es más cómodo trabajar con nombres.

También debe recordar que para facilitar el manejo de nombres de máquinas, se creó una estructura de dominios que permitían la jerarquización de dichas máquinas y de sus nombres, y que, a partir de aquí, para conseguir una localización óptima de las mismas, existen unos servidores de nombres, denominados DNS. Con la aparición de DHCP nos podemos preguntar si una máquina cambia de dirección IP cada vez que se inicia ¿cómo pueden los servidores actualizar su correspondencia "dirección IP-nombre de máquina a la misma velocidad que cambia las direcciones?.

Para resolver este problema se podrían dar tres soluciones:

- Las máquinas no deben tener nombres. Resulta muy tedioso manejarlas con números de direcciones.

- Las máquinas reciben un nombre a la vez de una dirección IP. Pero las máquinas recibirán nombres distintos a cada sesión lo cual no tiene gracia.

- Asignar un nombre fijo en cada máquina y actualizar los DNS de forma que se haga corresponder a estos nombres las direcciones que dinámicamente se les va

asignando a cada máquina; esta última solución exige una coordinación importante entre los servidores de nombres y los servidores de direcciones IP.

Las diversas casas comerciales están dando solución a
este problema.

2.7.5 Instalación del servicio DHCP. En caso de que en su instalación de Linux no haya agregado este paquete puede descargar una versión de este servicio (*DHCPd*) del sitio ftp://ftp.isc.org/isc/dhcp/dhcp-2.0.tar.gz, es el *DHCP* de Paul Vixie, copie el archivo descargado a un directorio a su gusto, por ejemplo al directorio */tmp*.

Nos desplazamos al directorio */tmp*, ejecutamos:

```
[root@lhost] / # tar -zxvf dhcp-2.0.tar.gz
```

Nos trasladamos al directorio de los fuentes creado, y ejecutamos:

```
[root@lhost] / # ./configure
```

Este comando va a detectar el Sistema Operativo que estas usando (Basado en Unix) y el kernel que usas.

Luego, ejecutamos:

```
[root@lhost] / # make
```

Al finalizar la compilación, ejecutamos:

```
 [root@lhost] / # make install
```

para copiar los binarios a directorios ejecutables.

2.7.6 Configuración del servicio DHCP. Existen dos formas de configurar nuestro servicio *DHCP*, ambos indican al fichero */etc/dhcpd.conf* como archivo de configuración. A continuación las dos formas de configurar el servicio:

- La primera forma es creando ó editando el fichero */etc/dhcpd.conf* con las siguientes líneas:

```
# dhcpd.conf - úselo a su propio riesgo
#
#determina el servidor de la red
option domain-name "sistemas.cutb.edu.co";
#determina la unbicación del servidor de nombres
option domain-name-servers ns.sistemas.cutb.edu.co;
#determina el servidor dhcp
server-name "dhcp.sistemas.cutb.edu.co ";
#determina la subred y la máscara de red a usar
subnet 192.168.0.0 netmask 255.255.255.0 {
#determina el rango de las direcciones IP privadas clase
#C a usar (solo para este caso)
range 192.168.0.1 192.168.0.5;
#
option routers 192.168.0.1;
option broadcast-address 192.168.0.255;
option subnet-mask 255.255.255.0;
#especifica la dirección IP del servidor
option domain-name-servers 192.168.0.1;
}
```

➢ La segunda forma es siguiendo una serie de pasos a describir, primero se verifica si el kernel soporta multicast con la instrucción *ifconfig –a*, en caso de que no lo soporte debemos recompilar el núcleo en la opción *networking options* y seleccionar *multicast*.

Luego añadimos el encaminamiento para la dirección de broadcast 255.255.255.255 (por compatibilidad con clientes como Windows, que no admiten la dirección de broadcast de la red local):

```
[root@lhost] / #route add -host 255.255.255.255 dev eth0
```

Configuramos las opciones DHCP, creando o editando el fichero */etc/dhcpd.conf*.

Por ejemplo:

```
#
#dhcpd.conf
#
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
```

```
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
#especifica el IP de los servidores
option domain-name-servers 192.168.1.1, 192.168.1.2;
#especifica el nombre del servidor
option domain-name "dhcp.sistemas.cutb.edu.co";
subnet 192.168.1.0 netmask 255.255.255.0 {
#los rangos de IP clase C a asignar
range 192.168.1.10 192.168.1.100;
range 192.168.1.150 192.168.1.200;
}
#especifica una dirección fija para el servidor haagen
host haagen {
hardware ethernet 08:00:2b:4c:59:23;
fixed-address 192.168.1.222;
}
#fin de archivo
```

Hecho esto ejecutamos el demonio */usr/sbin/dhcpd* en el arranque.

Para más información podemos consultar los manuales de ayuda de este servicio por medio de las páginas del man.

3. SERVICIO HTTP

Actualmente el servidor *HTTP*, más usado en la red es el Apache Web Server. Está demostrado que es el servidor más flexible, rápido y estable que se encuentra en el mercado, además tiene compenetración y es compatible con versiones del competidor más cercano el *NCSA*, debido a que es un servidor tipo *NCSA* por ende debe ser compatible con las versiones de este servidor.

Ahora bien, sigamos a la pregunta ¿qué es el Apache?. Apache salió de la palabra *A PatCHy sErver* y no de la tribu nativa norteamericana, la tribu apache.

Podemos concluir que el Apache es un parche que permite que nuestra máquina actúe como servidor HTTP, más no solo como eso apache es:

- Un poderoso, flexible y compatible servidor HTTP/1.1

- Implementa los últimos protocolos incluyendo HTTP/1.1 (RFC2616)

- Es fácil de configurar y posee gran cantidad de módulos externos para distintas operaciones.
- Puede ser configurable a gusto propio, escribiendo módulos con el Apache API

- Ejecuta en Windows NT/2000, Novell Netware, OS/2 y sistemas UNIX.

- Está en desarrollo constantemente eso permite más rápida actualización frente a los cambios.

- Por ser código abierto, actúa más rápido la reingeniería, aceptando reportes de BUGS, comentarios, módulos y parches.

Dadas a las tantas ventajas que posee el apache nos cuesta creer que sea gratuito, ¡Sí!, gratuito solamente se necesita leer su licencia dada en este link Licencia del Apache, y seguir sus indicaciones para uso y redistribución.

3.1 DISEÑO DEL SERVICIO HTTP CON APACHE WEB SERVER.

El diseño de un servidor HTTP, está delimitado a las exigencias que este tenga, a las políticas de la empresa y el uso que se le tenga que dar. Actualmente este

servicio es usado para mucho más que un protocolo de transferencia de hipertexto, sino que es usado para el intercambio de archivo, negocios por páginas, transacciones y quien sabe cuantas cosas más. Pero a pesar de todo esto el servicio debe ser administrado por mínimo dos personas, que detallaremos a continuación:

- Administrador del servicio: Es el responsable de la configuración del servidor, la ejecución y creación de CGI y scripts, seguridad de los archivos y la administración de usuarios.

- Web master: Es el encargado de administrar la información con el sitio, de actualizarlo, solventar los problemas con los usuarios, prácticamente es el mediador de relaciones entre los usuarios y el servicio.

Como habíamos dicho el servicio de HTTP a medida del tiempo aumentó funciones y propiedades haciendo que se modularan los servicios, permitiendo así que exista un administrador ó varios para cada uno de los servicios activos. El servidor debe constar de normas de seguridad tal como cualquier otro, al administrador de la red conjuntamente con el administrador del servicio deben establecer políticas de desempeño del servicio así como la interacción de este con

otros servicios prestados en la misma red. Igual que como otros servicios que hemos tratado la pericia en el manejo y uso que tenga el administrador ayudará al diseño del servicio.

El diseñador debe tener en cuenta los siguientes ítem para obtener conclusiones y resultados referentes al servicio:

- De que tipo se prestará el servicio (empresarial, educativo, etc.).

- ¿Cuál será la cobertura del servicio (LAN, Internet, MAN, WAN)?.

- Zona física que se encontrará el servicio.

- Número de usuarios que accederán al servicio (tipo de conexión, privilegio, flujo de información que tratará).

- Valor de la información.

- ¿Cómo podrá ser usado el servicio en el futuro?.

- ¿Cuánto crecerá el servicio en los próximos años?.

- Nivel de tecnología necesaria para el soporte.

De los datos anteriores el diseñador puede sacar conclusiones y especificar que módulos del servicio instalar y cuales no. Podrá especificar el nivel de seguridad físico y lógico.

3.2 INSTALACION DEL APACHE WEB SERVER.

3.2.1 La librería MM. Antes de comenzar en sí la instalación del Apache es importante instalar la biblioteca de funciones MM.

¿Qué es la biblioteca de funciones MM?, la biblioteca de funciones MM es una librería abstracta de segunda línea, que simplifica el uso de la memoria entre clonaciones, es extremadamente relacionada a los procesos bajo plataformas Unix.

En el primer nivel oculta a todas las plataformas dependiendo de los detalles de la implementación; locaciones y aseguramiento, cuando ocurren tratos con los

segmentos de memoria, y en el segundo nivel. provee un alto nivel de *malloc* estilo *API* para una forma conveniente y bueno conocemos la forma como trabaja con las estructuras de datos dentro de estos segmentos de memoria.

El uso de la librería es cobijada bajo los términos de un código abierto, estilo licencia BSD, ya que originalmente fue escrita para usarla dentro de las próximas versiones de Apache Web server como una librería base para proveer espacios en la memoria a los módulos de el Apache, actualmente los módulos gran cantidad de alocaciones de memoria, cuando no se cruzan con los preclonaciones de los procesos del servidor. El requerimiento actual viene de los módulos *mod_ssl*, *mod_perl* y *mod_php*, los cuales se benefician de una forma de fácil uso de los espacios de memoria.

Esta instalación asume:

- Comandos compatibles con Unix.

- El directorio fuente es */var/tmp*, otros directorios son posibles.

- Las instalaciones fueron probadas en versiones últimas de Red Hat.

- Todos los pasos en la instalación deben ser en la cuenta root.

El paquete puede descargarse de: http://www.engelschall.com/sw/mm/ Antes de descomprimir el archivo *tar*, es buena idea hacer una lista de archivos en el sistema antes de instalar MM, podemos hacerlo de esta manera:

```
find /* > MM1
```

Luego:

```
find /* > MM2
```

Instalamos el software y luego ejecutamos:

```
diff MM1 MM2 > MM-Installed
```

Para obtener una lista de lo que cambió.

Copiamos el archivo a un directorio temporal y lo descomprimimos.

```
[root@lhost /]# cp mm-version.tar.gz /var/tmp
[root@lhost /]# cd /var/tmp
[root@lhost ]/tmp# tar xzpf mm-version.tar.gz
```

Nos movemos ahora al directorio *mm* creado, y escribimos los siguientes comandos en nuestra terminal:

```
./configure \
--disable-shared \
--prefix=/usr
```

Esto esplica que MM debe configurarse para un hardware particular con:

```
Disable shared libraries
```

Ahora podemos compilar e instalar MM en el servidor:

```
[root@lhost ]/mm-version# make
[root@lhost ]/mm-version# make test
[root@lhost ]/mm-version# make install
```

El comando `make test` es para probar que se esta ejecutando correctamente de esta manera, esto lo realizamos antes de instalarlo.

Luego limpiamos los directorios y archivos que ya no necesitamos:

```
[root@lhost /]# cd /var/tmp
[root@lhost ]/tmp# rm -rf mm-version/ mm-version.tar.gz
```

3.2.2 Copia de ficheros. Luego de instalar la librería MM que podríamos necesitar podemos instalar el Apache, la instalación del apache comienza desde el punto de la descarga del código fuente ó su distribución binaria, sí ha descargado la binaria basta con instalar el paquete con:

```
rpm -i <nombre_del_rpm>
```

y listo en caso de descargar las fuentes, la instalación se deriva en tres pasos seleccionar los módulos a instalar, crear la configuración para nuestro sistema y compilar los ejecutables.

Aquí les mostraremos paso a paso la instalación con las fuentes descargadas:

Copiamos el archivo descargado al directorio temporal estos módulos son opcionales dependen de lo que se desee instalar

```
[root@lhost ]/# cp apache_version.tar.gz /var/tmp
```

Copiamos el modulo de secure socket layer al directorio temporal

```
[root@lhost ]/# cp mod_ssl-version-version.tar.gz /var/tmp
```

Copiamos el modulo de perl al directorio temporal

```
[root@lhost ]/# cp mod_perl-version.tar.gz /var/tmp
```

Copiamos el modulo de php al directorio temporal

```
[root@lhost ]/# cp php-version.tar.gz /var/tmp
```

Nos dirigimos al directorio temporal extendemos nuestros archivos tar:

```
[root@lhost ]/# cd /var/tmp/
[root@lhost ]/tmp# tar xzpf apache_version.tar.gz
[root@lhost ]/tmp# tar xzpf mod_ssl-version-version.tar.gz
[root@lhost ]/tmp# tar xzpf mod_perl-version.tar.gz
[root@lhost ]/tmp# tar xzpf php-version.tar.gz
```

Como el Apache Web Server no puede ejecutarse desde el usuario root, por obvias razones crearemos un usuario:

```
[root@lhost ]/# useradd -c "Apache Server" -u 80 -s
/bin/false -r -d /home/httpd www 2>/dev/null || :
```

Es necesario advertir que usamos como servidor lhost, este nombre puede variar dependiendo del nombre que se le haya dado a la máquina.

Antes de instalar el Apache debemos antes instalar los módulos, que elegimos.

3.2.3 Instalación del módulo SSL. Ahora nos desplazamos al nuevo directorio:

```
/var/temp/mod_ssl-version-version
```

Desde allí ejecutamos el siguiente Script desde nuestra terminal:

```
CC="egcs" \
CFLAGS="-O9 –funroll -loops –ffast -math –malign -double -
mcpu=
pentiumpro -march=pentiumpro –fomit –frame -pointer -fno-
exceptions" \
./configure \
--with-apache=../apache_1.3.18 \
--with-crt=/etc/ssl/certs/server.crt \
--with-key=/etc/ssl/private/server.key
```

- La directiva --with-apache=../apache_1.3.18 indica el directorio donde se encuentra el fuente del apache, en este caso es la versión 1.3.18.

- La directiva --with-crt=/etc/ssl/certs/server.crt indica donde se encontrará la clave pública para la encripción.

- La directiva --with-key=/etc/ssl/private/server.key indica donde se encontrará la clave privada para la encripción.

Importante: El servidor *OpenSSL* debe estar instalado para la ejecución de este módulo.

Advertencia: La ejecución de un servidor con *SSL* es sometida a leyes internacionales, el uso de servidores con esta características están restringidos para países fuera de los Estados Unidos de Norteamérica.

Luego se debe indicar el parámetro de máximo número de clientes normalmente el Apache conserva 256, sí queremos aumentar su desempeño debemos dirigirnos al directorio de los fuentes del Apache */var/temp/apache_version* y editamos el archivo *httpd.h* y cambiamos los números del final de la línea `#define HARD_SERVER_LIMIT 256` por el máximo número de clientes que nosotros queramos.

3.2.4 Instalación del módulo PHP4. Sí nosotros deseamos instalar el módulo para Apache PHP4, este módulo servirá para que nuestro servidor web ejecute comandos PHP4.

Para instalar nos desplazamos hasta el directorio de los fuentes del Apache, y desde la terminal escribimos:

```
CC="egcs" \
OPTIM="-O9 -funroll-loops -ffast-math -malign-double -mcpu=
pentiumpro   -march=pentiumpro   -fomit-frame-pointer   -fno-
exceptions"
\
CFLAGS="-DDYNAMIC_MODULE_LIMIT=0" \
./configure \
--prefix=/home/httpd \
--bindir=/usr/bin \
--sbindir=/usr/sbin \
--libexecdir=/usr/lib/apache \
--includedir=/usr/include/apache \
--sysconfdir=/etc/httpd/conf \
--localstatedir=/var \
--runtimedir=/var/run \
--logfiledir=/var/log/httpd \
--datadir=/home/httpd \
--proxycachedir=/var/cache/httpd \
```

```
--mandir=/usr/man
```

Usted debe notar que el comando `-DDYNAMIC_MODULE_LIMIT=0` esta opción se deshabilita para que abra los módulos dinámicamente durante la compilación del Apache.

Ahora nos desplazamos a nuestro directorio con los fuentes del módulo de *PHP4*, editamos el archivo *php_pgsql.h* desde nuestra terminal, ubicado en *../ext/pgsql/php_pgsql.h* y cambiamos las líneas:

```
#include libpq-fe.h
#include libpq/libpq-fs.h
```

Por:

```
#include /usr/include/pgsql/libpq-fe.h
#include /usr/include/pgsql/libpq/libpq-fs.h
```

Esto lo hacemos para indicar la locación de las librerías usadas por *PHP4* de la base de datos *PostgressSQL*. En los sistemas Linux Red Hat se encuentran en */usr/include/pgsql* esta ubicación puede cambiar de acuerdo al sistema.

Ahora configuramos e instalamos *PHP4* en nuestro servidor Linux desde la terminal:

```
CC="egcs" \
CFLAGS="-O9 -funroll-loops -ffast-math -malign-double -mcpu=
pentiumpro   -march=pentiumpro   -fomit-frame-pointer   -fno-
exceptions
-I/usr/include/openssl" \
./configure \
--prefix=/usr \.90
--with-exec-dir=/usr/bin \
--with-apache=../apache_1.3.18 \
--with-config-file-path=/etc/httpd \
--disable-debug \
--enable-safe-mode \
--with-imap \
--with-ldap \
--with-pgsql \
--with-mm \
--enable-inline-optimization \
```

```
--with-gnu-ld \
--enable-memory-limit
```

Las directivas `--with-imap`, sirven sí desea soporte para *IMAP & POP*, `--with-ldap` sí desea soporte para *LDAP*, `--with-pgsql` sí desea soporte para la base de datos *PostgressSQL* .

La configuración hecha contiene:

- Compilación sin símbolos de depuración.
- Activo modo seguro por defecto.
- Incluye soporte IMAP y POP.
- Incluye soporte a los directorios LDAP.
- Incluye soporte a la base de datos PostgressSQL.
- Incluye soporte mm que incrementa el rendimiento de las librerías en memoria.

- Activo la optimización en línea para mejor desempeño.

- Soporte para compilación con límite de memoria.

- Asume el compilador C usado por *GNU*.

Luego desde el terminal ejecutamos los comandos para instalar:

```
[root@lhost ]/php-4.0# make
[root@lhost ]/php-4.0# make install
```

3.2.5 Instalación del módulo Perl. Para agregar nuestro módulo a nuestro servidor apache, debemos crear su instalador. Como primera instancia nos movemos a nuestro directorio de los fuentes de Perl *../Perl-version/* y desde la terminal ejecutamos los comandos:

```
perl Makefile.PL \
EVERYTHING=1 \
APACHE_SRC=../apache_1.3.12/src \
```

```
USE_APACI=1 \
PREP_HTTPD=1 \
DO_HTTPD=1.91

[root@lhost ]/mod_perl-version# make
[root@lhost ]/mod_perl-version# make install
```

3.2.6 Configuración de los archivos de instalación para la inserción de módulos. Luego de haber incluido los módulos para soporte y uso debemos desplazarnos a nuestro directorio de fuentes del Apache *../apache_version/* y ejecutamos el siguiente script:

```
SSL_BASE=SYSTEM \
EAPI_MM=SYSTEM \
CC="egcs" \
OPTIM="-O9 -funroll-loops -ffast-math -malign-double -mcpu=
pentiumpro   -march=pentiumpro   -fomit-frame-pointer   -fno-
exceptions"
\
CFLAGS="-DDYNAMIC_MODULE_LIMIT=0" \
```

```
./configure \
--prefix=/home/httpd \
--bindir=/usr/bin \
--sbindir=/usr/sbin \
--libexecdir=/usr/lib/apache \
--includedir=/usr/include/apache \
--sysconfdir=/etc/httpd/conf \
--localstatedir=/var \
--runtimedir=/var/run \
--logfiledir=/var/log/httpd \
--datadir=/home/httpd \
--proxycachedir=/var/cache/httpd \
--mandir=/usr/man \
--add-module=src/modules/experimental/mod_mmap_static.c \
--add-module=src/modules/standard/mod_auth_db.c \
--enable-module=ssl \
--enable-rule=SSL_SDBM \
--disable-rule=SSL_COMPAT \
--activate-module=src/modules/php4/libphp4.a \
--enable-module=php4 \
--activate-module=src/modules/perl/libperl.a \
```

```
--enable-module=perl \
--disable-module=status \
--disable-module=userdir \
--disable-module=negotiation \
--disable-module=autoindex \.92
--disable-module=asis \
--disable-module=imap \
--disable-module=env \
--disable-module=actions
```

Donde:

- La primera línea es requerida si agregamos el módulo *SSL* a nuestro servidor.

- La segunda línea es requerida si usamos las librerías en la memoria para el apache.

- La línea `–add`.

- `-module=src/modules/experimental/mod_mmap_static.c \` sí nosotros tenemos la intención de usar el módulo mod_mmap, que nos servirá para la optimización de nuestro servidor apache, eso lo veremos más adelante.

- La línea `--add-module=src/modules/standard/mod_auth_db.c \` sí nosotros tenemos la intención de usar el módulo *mod_auth_db*, que nos servirá para la seguridad de las bases de datos que pueda usar nuestro servidor Web.

- Las `líneas --enable-module=ssl \`
`--enable-rule=SSL_SDBM \`
`--disable-rule=SSL_COMPAT \` son requeridas si nosotros tenemos incluido soporte para el módulo *SSL* para la encripción de datos.

- Las líneas `--activate-module=src/modules/php4/libphp4.a \`
`--enable-module=php4 \` son requeridas si nosotros incluimos soporte para *PHP4* con el módulo *PHP4*.

- Las líneas `--activate-module=src/modules/perl/libperl.a \`
`--enable-module=perl \` son requeridas sí nosotros incluimos el módulo de Perl a nuestro servidor.

Esta configuración de apache contiene:

- El módulo *mod_mmap* para mejorar desempeño.

- El módulo *mod_auth_db* para la autenticación de contraseñas de usuarios.

- El módulo *mod_ssl* para la encripción de datos y seguridad en la comunicación.

- El módulo *mod_php4* para *php server-side scripting language* y poder cargar páginas con PHP.

- El módulo *mod_perl* para mejor seguridad y desempeño que los *scripts cgi* por defecto.

- Desactivo el módulo de estado.

- Desactivo el módulo de directorio de usuario.

- Desactivo el módulo de negociación.

- Desactivo el módulo de autoíndice.

- Desactivo el módulo asis.

- Desactivo el módulo IMAP.

- Desactivo el módulo env.

- Desactivo el módulo actions.

Nota: Es importante remover todos los módulo durante el tiempo de configuración podría afectar el desempeño de su Apache Web Server. En su configuración, deberá remover los módulos menos usados, ya que estos afectan el desempeño y limita los riesgos en la seguridad de nuestro servidor.

3.2.7 Instalando el Apache y sus módulos. Luego de la inserción de los módulos que deseamos instalar, podemos instalar el Apache de esta manera cerciorémonos de estar en el directorio de los fuentes del Apache, *../apache_versión/* y ejecutamos los siguientes comandos:

```
[root@lhost ]/apache_1.3.18# make
[root@lhost ]/apache_1.3.18# make install
[root@lhost ]/apache_1.3.18# rm -f /usr/sbin/apachectl
[root@lhost ]/apache_1.3.18# rm -f /usr/man/man8/apachectl.8
[root@lhost ]/apache_1.3.18# rm -rf /home/httpd/icons/
[root@lhost ]/apache_1.3.18# rm -rf /home/httpd/htdocs/
```

Sí habíamos construido el instalador para el módulo de *php4* lo debemos ejecutar ahora desplazándonos al directorio fuente del *php4* y ejecutar el código siguiente:

```
[root@lhost ]/apache_1.3.18# cd /var/tmp/php-4.0
[root@lhost   ]/php-4.0.0#   install   -m   644   php.ini.dist
/usr/lib/php.ini
[root@lhost ]/php-4.0.0# rm -rf /etc/httpd/conf/ssl.crl/
[root@lhost ]/php-4.0.0# rm -rf /etc/httpd/conf/ssl.crt/
[root@lhost ]/php-4.0.0# rm -rf /etc/httpd/conf/ssl.csr/
[root@lhost ]/php-4.0.0# rm -rf /etc/httpd/conf/ssl.key/
[root@lhost ]/php-4.0.0# rm -rf /etc/httpd/conf/ssl.prm/
[root@lhost ]/php-4.0.0# rm -f /etc/httpd/conf/srm.conf
srm.conf.default access.conf access.conf.default
```

Luego debemos limpiar los archivos temporales que nos sirvieron para la instalación y configuración, podemos ejecutar los comandos:

```
[root@lhost ]/# cd /var/tmp
[root@lhost ]/tmp# rm -rf apache-version/ apache-version tar.gz mod_ssl-version-version/ mod_ssl-version-version tar.gz php-version/ php-version.tar.gz mod_perl-version/ mod_perl-version.tar.gz
```

De esta manera ya tenemos instalado nuestro servidor Web Apache en nuestro equipo Linux, a continuación mostraremos los pasos inmediatos a una instalación Apache.

Los archivos de configuración para los diferentes servicios son muy específicos dependiendo de las necesidades, y la arquitectura de la red. Algunos pueden instalar el Apache Web Server solamente para mostrar páginas Web, otros pueden instalarlo para la conexión a una base de datos y e-commerce con soporte *SSL*, etc.. Nosotros trataremos de mostrar las diferentes posibilidades de servicios que nos ofrece el Apache.

Nosotros nos enfocaremos en la optimización y seguridad de los archivos, dejando atrás todas las especificaciones y ajustes personales. Usted tendrá la necesidad de leer la documentación que viene con los programas, y esperamos que la entienda.

Para ejecutar el servidor Apache, se requieren los siguientes archivos y ubicarlos en los directorios:

- Archivo *httpd.conf* en el directorio */etc/httpd/conf/*

- Archivo *apache* en el directorio */etc/logrotate.d/*

- Archivo *httpd* script en el directorio */etc/rc.d/init.d/*

3.3 CONFIGURACION DEL APACHE.

En este aparte trataremos la configuración del Apache Web Server, a pesar de que existen utilidades en Linux para la configuración de este servicio tal como lo es el linuxconf y la herramienta visual que provee nuestro gestor de ventana hemos decidido explicar la configuración mediante los archivos de configuración

que se instala en nuestro servidor. Cómo la mayoría de los servicios en Linux se configura por medios de archivos con algunas sentencias y directivas.

3.3.1 Configurando el archivo /etc/httpd/conf/httpd.conf. En este archivo se encuentra la configuración principal de nuestro servidor Web. Para su configuración Apache dispone de una gran variedad de opciones, a lo cual es importante leer la documentación que viene con el Apache para conocer cada una de las directivas. A continuación presentaremos una configuración sencilla de este archivo introducido el módulo *SSL*:

```
# Seccion 1: Configuración Global
#
ServerType standalone
#Esta directiva especifica como el Apache correrá en nuestro
#sistema. Usted #puede elegir si correrá bajo el super.95
#servidor inetd, ó correrá solo. Para mejor #desempeño es
#necesario colocarlo "solo" standalone
ServerRoot "/etc/httpd"
#Esta directiva indica el directorio donde se encuentran los
#archivos de configuración del Apache, es usado por el apache
```

```
#para conocer donde se #encuentran los archivos cada vez que
#se inicia.
PidFile /var/run/httpd.pid
#Esta directiva indica donde el sistema almacena el proceso
#demonio cuando es ejecutado, #solo es requerida cuando el
#Apache se ejecuta en modo standalone
ResourceConfig /dev/null
#Esta directiva indica al Apache donde se encuentra el
#antiguo archivo srm.conf que el #Apache lee luego de leer el
#archivo httpd.conf, cuando nosotros indicamos /dev/null
#solamente usamos el archivo httpd.conf de esta manera es más
#fácil manejar la configuración.
AccessConfig /dev/null
###Esta directiva indica la ubicación del archivo
#access.conf, que el Apache lee luego de leer el archivo
#srm.conf, al colocar /dev/null especificamos que no lo lea,
#de esa manera para la configuración manejaremos un solo
#archivo httpd.conf
Timeout 300
###Esta opción indica el tiempo que esperará el Apache para
#las peticiones POST, GET, PUT y las transmiciones ACKs.
```

```
KeepAlive On
###Esta opción indica la persistencia de conecciones con
#nuestro servidor. Para un mejor desempeño es necesario
#tenerlo en On.
MaxKeepAliveRequests 0
###Esta opción especifica el número de peticiones mantenidas
#por conexión, cuando la opción KeepAlive esta en On. Cuando
#el valor de esta opción es 0, existe ilimitadas peticiones
#en nuestro servidor, para mejor desempeño dejarlo en 0.
KeepAliveTimeout 15
#Esta opción especifica el tiempo en segundos, que el Apache
#esperará por una petición subsecuente antes de cerrar la
#conexión. 15 segundos es un buen tiempo para la espera en un
#servidor.
MinSpareServers 16
#Esta opción especifica el mínimo número de procesos hijos
#activos en nuestro servidor para Apache, cuando no hay a la
#mano ninguna petición. Esta es una de las opciones
#importantes para mantener un buen desempeño del Apache. para
#un máximo de operación, un valor de 16 es recomendado para
#varias ramas en el Internet..96
```

```
MaxSpareServers 64
#Si la antigua opción se refería al mínimo de procesos
#hijos activos, ésta indica el máximo número de procesos
#hijos activos. El valor de 64 es recomendado.
StartServers 16
#Esta opción especifica el número de procesos hijos que
#serán creados por el Apache al iniciar su servicio. Esto es
#muy importante nuevamente para mantener el desempeño de
#nuestro servidor. 16 es un buen número.
MaxClients 512
#Esta opcion especifica el número máximo de peticiones
#simultaneas que soportará el Apache. Es importante conocer
#el número de clientes que atenderá nuestro servidor y la
#seguridad que se debe tener con los archivos de este.
### 512 clientes es un buen número para un servidor Internet
#común, pero si sabe que accederán menos clientes haga esta
#formula MAX_CLIEN= NUM_CLIEN + (NUM_CLIEN/3)
MaxRequestsPerChild 100000
# Esta opción especifica el número máximo de peticiones que
#mantendrá nuestro servidor por cada proceso hijo de nuestro
#servidor.
```

```
### Seccion 2: Configuración Principal de nuestro servidor
#
Port 80 <IfDefine SSL>
Listen 80
Listen 443
</IfDefine>
User www
###Esta opción especifica el UID (user identified) que
#nuestro servidor Apache usará para ejecutarse. Esto es
#importante para crear usuarios con un mínimo acceso a el
#sistema y funciones, solamente podrá ejecutar el demonio de
#nuestro servidor Web.
Group www
###Esta opción especifica el GID (Group Identified) que
#nuestro servidor Apache usará para ejecutarse. Esto es
#importante para la creación de nuevos grupos con acceso
#mínimo al sistema y funciones, sólo con el propósito de
#ejecutar el demonio de nuestro servidor Web.
ServerAdmin admin@lhost.com
ServerName www.lhost.com
DocumentRoot "/home/httpd/html" <Directory />
```

```
Options None
AllowOverride None
Order deny,allow
Deny from all
```

3.3.2 Configuración del archivo /etc/logorotate.d/apache. Luego de la configuración de nuestro archivo *httpd.conf*, podemos configurar el archivo apache que se encuentra en el directorio */etc/logrotate.d/* , en caso de no existir puede crearlo con el siguiente comando:

```
touch /etc/logrotate.d/apache
```

Este archivo nos servirá para rotar los archivos *log* cada semana. El archivo debe contener las líneas siguientes:

```
/var/log/httpd/access_log {
missingok
postrotate
/usr/bin/killall -HUP httpd
endscript
```

```
}
/var/log/httpd/error_log {
missingok
postrotate
97
/usr/bin/killall -HUP httpd
endscript
}
/var/log/httpd/ssl_request_log {
missingok
postrotate
/usr/bin/killall -HUP httpd
endscript
}
/var/log/httpd/ssl_engine_log {
missingok
postrotate
/usr/bin/killall -HUP httpd
endscript
}
```

Estas líneas automáticamente rotarán los logs de *ssl_request* y *ssl_engine*. Si se pretende ejecutar el Apache sin *SSL* debemos suprimir las líneas a que está referido. Los archivos de log se encontrarán en el directorio */var/log/httpd/* cada uno con el nombre del script anterior, si queremos cambiar de directorio no hay problema.

3.3.3 Configuración del archivo /etc/rc.d/init.d/httpd. Hasta ahora hemos hecho los archivos de configuración y configuramos la salida de los log, ahora crearemos un script que nos servirá para iniciar nuestro servicio al iniciar la máquina. Este script debe encontrarse en el directorio */etc/rc.d/init.d/* con el nombre *httpd*.

El script si no se encuentra podemos crearlo con el comando:

```
touch /etc/rc.d/init.d/httpd
```

Este script debe contener:

```
#!/bin/sh
#
# Script para ejecutar Apache Web Server
```

```
#
# chkconfig: 345 85 15
# description: Apache es un servidor WWW. Es usado para
servir \
# Archivos HTML y CGI.
# processname: httpd
# pidfile: /var/run/httpd.pid
# config: /etc/httpd/conf/httpd.conf # Fuente de la libreria
de funciones.
. /etc/rc.d/init.d/functions # Vea como le hemos llamado.
case "$1" in
start)
echo -n "Starting httpd: "
daemon httpd -DSSL
echo
touch /var/lock/subsys/httpd
;;
stop)
echo -n "Shutting down http: "
killproc httpd
echo
```

```
rm -f /var/lock/subsys/httpd
rm -f /var/run/httpd.pid
;;
status)
status httpd
;;
restart)
$0 stop
$0 start
;;
reload)
echo -n "Reloading httpd: "
killproc httpd -HUP
echo
;;
*)
echo "Usage: $0 {start|stop|restart|reload|status}"
exit 1
esac exit 0
```

Luego de crear ó editar el Script, debemos hacerlo ejecutable y cambiarle los permisos por defectos:

```
[root@lhost ]/# chmod 700 /etc/rc.d/init.d/httpd
```

Creamos un link simbólico para el Apache:

```
[root@lhost ]/# chkconfig --add httpd
```

Iniciamos nuestro servicio con:

```
[root@lhost ]/# /etc/rc.d/init.d/httpd start
```

A lo cual debe aparecer:

```
Starting httpd: [ OK ]
```

Para saber si nuestro servidor se está ejecutando podemos ejecutar el comando:

```
ps -aux | grep
```

Verá una serie de procesos *httpd* ejecutándose. ¿Porqué?, la razón es que a diferencia de otros servidores que usan un modelo único de servidor que se clonaba cada vez que se daba una petición y el servidor original volvería al puerto a esperar la otra petición. A pesar del diseño sencillo y robusto, la clonación (que en lenguaje UNIX se denomina *forking*) es una operación costosa para el sistema ya que con cargar más de dos conexiones resulta nefasto hasta para el hardware más robusto. Es difícil también llevar a cabo cualquier tipo de control de admisión que redujera el número de clonaciones que tenían lugar. Cuando el número de clonaciones es elevado, el servidor original tenía dificultad para saber cuántos de ellos se encontraban funcionando. Por ende los servidores deben por falta de recursos rechazar ó demorar conexiones. Apache a diferencia de estos servidores utiliza un grupo permanente de hijos ejecutándose en paralelo. Un proceso padre coordina a los hijos, este indica el número de hijos supervivientes, engendra nuevos hijos si es necesario e incluso acaba, en función con los hijos viejos si hay demasiados desocupados.

3.3.4 Inserción de los módulos en la configuración. Sí nuestra intención es usar el soporte de lenguaje PHP4 server side con nuestro servidor Web no olvidemos de agregar las siguientes líneas al archivo de configuración

/etc/httpd/conf/httpd.conf. Al editarlo debemos incluir las líneas siguientes entre los *tags*:

```
<IfModule mod_mime.c> y </IfModule>:
AddType application/x-httpd-php .php
AddType application/x-httpd-php .php3
AddType application/x-httpd-php-source .phps
```

Debemos luego reiniciar nuestro servicio de la siguiente manera:

```
[root@lhost ]/# /etc/rc.d/init.d/httpd restart
```

A lo cual debe aparecer:

```
Shutting down http: [ OK ]
Starting httpd: [ OK ]
```

Luego debemos testear nuestro nuevo módulo ejecutando un pequeño script en *PHP*, creamos un documento llamado *php.php* y lo localizamos en nuestro directorio raíz de nuestro servidor. El archivo debe contener las siguientes líneas:

```
<body bgcolor="#FFFFFF">.100
<?php phpinfo()?>
</body>
```

Sí usted tiene experiencia con el *PHP*, puede agregar ó hacer un script, bueno sí el módulo fue agregado al acceder al archivo desde nuestro explorador de la siguiente manera *http://my_web_server/php.php* debe aparecer en pantalla información acerca de nuestro servidor Linux.

Siguiente si desea que su servidor Web soporte la programación en Perl, pueda ser que este interesado en instalar un pequeño módulo perl llamado *Devel::Sysdump* que lo puede descargar de la página http://www.perl.com/CPAN/modules/by-module/Devel/, asegúrese de descargar el archivo *Devel-Symdump-2_00_tar.gz* en su defecto la versión más reciente.

La instalación de este módulo comienza copiando los archivos a un directorio temporal y descomprimiendolo:

```
[root@lhost ]/# cp Devel-Symdump-version.tar.gz /var/tmp/
[root@lhost ]/# cd /var/tmp/
[root@lhost ]/tmp# tar xzpf Devel-Symdump-version.tar.gz
```

Nos desplazamos a nuestro directorio temporal y ejecutamos los siguientes comandos para instalarlo:

```
[root@lhost ]/Devel-Symdump-2.00# perl Makefile.PL
[root@lhost ]/Devel-Symdump-2.00# make
[root@lhost ]/Devel-Symdump-2.00# make test
[root@lhost ]/Devel-Symdump-2.00# make install
```

Luego de la instalación del módulo en su sistema, usted deberá incluir en su archivo */etc/httpd/conf/httpd.conf* las siguientes líneas para ver la estadística de los diferentes módulo de Perl en nuestro servidor:

```
<Location /perl-status>
SetHandler perl-script
PerlHandler Apache::Status
Order deny,allow
Deny from all ###Negado para todos
Allow from 192.168.1.0/24 ###Solo puede acceder servidor IP
                          ###192.168.1.0 hasta el 24
</Location>
```

Usted debe reiniciar el servicio Apache para que los cambios surjan efecto:

```
[root@lhost ]/# /etc/rc.d/init.d/httpd restart
Shutting down http: [ OK ]
Starting httpd: [ OK ]
```

Finalmente debemos probar nuestro módulo *Devel-Sysdump*, para observar el estado de nuestro módulo de Perl.

Para verificar que esta trabajando abrimos desde nuestro Browser la dirección *http://my-web-server/perl-status/* . Donde *my_web_server* es la dirección de su servidor Web Apache.

Luego limpiamos nuestros directorios temporales:

```
[root@lhost ]/# cd /var/tmp
[root@lhost ]/tmp# rm -rf Devel-Symdump.version/ Devel-Symdump-version.tar.gz
```

La *CGI.pm* es una libreria Perl para el procesamiento de *CGI*, con la instalación debe venir una versión de esta libreria, es recomendable actualizarla en caso de

no ser la última versión, usted puede dirigirse a la página principal http://stein.cshl.org/WWW/software/CGI/cgi_docs.html y descargar el archivo *CGI_pm_tar.gz*. Para verificar la versión que se esté usando introducimos los comandos:

```
[root@lhost ]/# perl -e 'use CGI; print $CGI::VERSION."\n";'
```

Si ejecuta debe aparecer en pantalla la versión que estemos usando.

Para actualizar, primero debemos copiar el archivo descargado y descomprimirlo en un directorio temporal:

```
[root@lhost ]/# cp CGI_pm_tar.gz /var/tmp/
[root@lhost ]/# cd /var/tmp/
[root@lhost ]/tmp# tar xzpf CGI_pm_tar.gz
```

Nos movemos a el directorio creado y escribimos lo siguiente en la terminal para compilar e instalar las nuevas líbrerias en nuestro servidor Linux:

```
[root@lhost ]/CGI.pm-2.56# perl Makefile.PL
[root@lhost ]/CGI.pm-2.56# make
```

```
[root@lhost ]/CGI.pm-2.56# make test
[root@lhost ]/CGI.pm-2.56# make install
```

Luego debemos remover los directorios temporales:

```
[root@lhost ]/# cd /var/tmp
[root@lhost ]/tmp# rm -rf CGI.pm-version/ CGI_pm_tar.gz
```

3.4 MÓDULOS Y DIRECTIVAS EN EL APACHE.

El Apache por su desarrollo en paralelo tiene en su instalación módulos que permiten que este servidor realice ciertas funciones, cada módulo contiene cierto número de directivas que se ingresan en su configuración para la realización de ciertas funciones. Existen módulos que deben ser instalados y configurados aparte como *mod_ssl* y *mod_perl*, debido a que son colaboraciones de terceros y no se encuentran en el paquete base del *Apache Web Server*.

Usted puede crear sus propios módulos para optimizar ó personalizar su servidor.

A continuación presentaremos los módulos que contiene la versión *Apache 1.3.19*

con sus respectivas directivas.

3.4.1 Módulo del núcleo. Este módulo es el núcleo central de nuestro servidor Apache y siempre estará disponible al igual que sus directivas.

- Directiva AccessFileName

Sintaxis : AccessFileName nombre_archivo [nombre_archivo] ...

Valor por defecto: AccessFileName .htaccess

- Directiva AddDefaultCharset

Sintaxis: AddDefaultCharset On|Off|charset

Valor por defecto: AddDefaultCharset Off

- Directiva AddModule

Sintaxis: AddModule modulo [modulo] ...

- Directiva AllowOverride

Sintaxis: AllowOverride All|None|tipo_directiva [tipo_directiva] ...

Valor por defecto: AllowOverride All

El tipo_directiva puede ser uno de estos grupos:

- AuthConfig

Para el uso de autorización de directivas (AuthDBMGroupFile, AuthDBMUserFile, AuthGroupFile, AuthName, AuthType, AuthUserFile, Require, etc.).

- FileInfo

Para el control de tipos de documentos (AddEncoding, AddLanguage, AddType, DefaultType, ErrorDocument, LanguagePriority, etc.).

- Indexes

Para controlar el directorio indice de los documentos (AddDescription, AddIcon, AddIconByEncoding, AddIconByType, DefaultIcon, DirectoryIndex, FancyIndexing, HeaderName, IndexIgnore, IndexOptions, ReadmeName, etc.).

- Limit

Para controlar el acceso de servidores.

- Options

Para controlar las propiedades de algunos tipos de directorios. (Options and XBitHack).

- Directiva AuthName

Sintaxis: AuthName auth-domain.103

- Directiva AuthType

Sintaxis: AuthType Basic|Digest

- Directiva ClearModuleList

Sintaxis: ClearModuleList

- Directiva ContentDigest

Sintaxis: ContentDigest on|off

Valor por defecto: ContentDigest off

- Directiva DefaultType

Sintaxis: DefaultType MIME-type

Valor por defecto: DefaultType text/html

- Directiva <Directory>

Sintaxis: <Directory directory>

... </Directory>

- Directiva <DirectoryMatch>

Sintaxis: <DirectoryMatch regex>

... </DirectoryMatch>

- Directiva DocumentRoot

Sintaxis: DocumentRoot directory-filename

Valor por defecto: DocumentRoot /usr/local/apache/htdocs

- Directiva ErrorDocument

Sintaxis: ErrorDocument error-code document

- Directiva ErrorLog

Sintaxis: ErrorLog filename|syslog[:facility]

3.4.2 Módulos de multiprocesamiento (MPM). Los módulos de multiprocesamiento implementan un híbrido de multiprocesamiento y servidor multi hilos, con un número variable de hilos por proceso.

Un sencillo control de procesos es activado cada vez que un proceso hijo indicado por las directivas *NumServers* al inicio del servicio. Cada proceso hijo crea hilos especificado por la directiva *StartThreads*.

Apache siempre intenta tener cierto número de hilos esperando alguna petición, de esta manera el cliente no debe esperar hasta que se cree un nuevo hilo. Para cada proceso hijo, Apache accede para la creación ó destrucción de hilos para mantener un número de hilos para ese proceso, esta cantidad esta especificada en la directiva *MinSpareThreads* y *MaxSpareThreads*. Los módulos son:

- Nombre: dexter.

Fuente: dexter.c

Identificador: mpm_dexter_module

Directivas: ConnectionStatus, CoreDumpDirectory, Group, PidFile, Listen, ListenBacklog, LockFile, MaxRequestsPerChild, MaxSpareThreads, MaxThreadsPerChild, MinSpareThreads, NumServers, ScoreBoardFile, SendBufferSize, StartThreads, User

- Nombre: mpmt_pthread

Fuente: mpmt_pthread.c

Identificador: mpm_mpmt_pthread_module

Directivas: CoreDumpDirectory, Group, PidFile, Listen, ListenBacklog, LockFile, MaxClients, MaxRequestsPerChild, MaxSpareThreads, MinSpareThreads, ScoreBoardFile, SendBufferSize, StartServers, ThreadsPerChild, User

- Nombre: perchild

Fuente: perchild.c

Identificador: mpm_perchild_module

Directivas: ConnectionStatus, CoreDumpDirectory, Group, PidFile, Listen, ListenBacklog, LockFile, MaxRequestsPerChild, MaxSpareThreads, MaxThreadsPerChild, MinSpareThreads, NumServers, ScoreBoardFile, SendBufferSize, StartThreads, User

Directiva: AssignUserID

Sintaxis: Pone en un servidor virtual al proceso hijo especificado.

Directiva: ChilePerUserID

Sintaxis: Especifica un usuario y grupo para cada proceso hijo.

- Nombre: prefork

Fuente: prefork.c

Identificador: mpm_prefork_module

Directivas: CoreDumpDirectory, Group, PidFile, Listen, ListenBacklog, LockFile, MaxClients, MaxRequestsPerChild, MaxSpareServers, MinSpareServers, ScoreBoardFile, SendBufferSize, StartServers, User.

- Nombre: MaxSpareServers

Sintaxis: MaxSpareServers number

Valor por defecto: MaxSpareServers 10

- Nombre: MinSpareServers

Sintaxis: MinSpareServers number

Valor por defecto: MinSpareServers 5

Presentamos ahora las directivas comunes que comparten los módulos de multiprocesamiento.

- Directiva: ConnectionStatus

Sintaxis: ConnectionStatus on|off

Valor por defecto: ConnectionStatus on

- Directiva: CoreDumpDirectory

Sintaxis: CoreDumpDirectory directory

Valor por defecto: the same location as ServerRoot

- Directiva: Group

Sintaxis: Group unix-group

Valor por defecto: Group #-1

- Directiva: PidFile

Sintaxis: PidFile filename

Valor por defecto: PidFile logs/httpd.pid

- Directiva: Listen

Sintaxis: Listen [IP-address:]port number

- Directiva: ListenBacklog

Sintaxis: ListenBacklog backlog

Valor por defecto: ListenBacklog 511

- Directiva: LockFile

Sintaxis: LockFile filename

Valor por defecto: LockFile logs/accept.lock

- Directiva: MaxClients

Sintaxis: MaxClients number

Valor por defecto: MaxClients 8 (with threads)

Apache aparte a los módulos tratados cuenta con cientos de módulos más el cual incrementa su capacidad, desempeño y eficacia, para más información acerca de los módulos visite la página http://www.apache.org/.

3.5 Seguridad completa en el Apache.

A continuación presentaremos cierto grupo de políticas que se pueden implementar para asegurar nuestro servidor Apache.

3.5.1 Asegurando Archivos y Directorios. Para garantizar la seguridad de nuestro servidor *Apache* es necesario cambiar los permisos de algunos archivos y directorios importantes, he aquí una lista de archivos y directorios que tienen muchos permisos por defecto.

El programa binario *httpd* puede ser puesto como solo lectura para el superusuario root, y ejecutado por el propietario, grupo y otros para mejor seguridad. Los directorios */etc/httpd/conf* y */var/log/httpd* no necesitan ser leídos, escritos ó ejecutados por otros usuarios. He aquí los comandos al shell para esto:

```
[root@lhost ]/# chmod 511 /usr/sbin/httpd
[root@lhost ]/# chmod 750 /etc/httpd/conf/
[root@lhost ]/# chmod 750 /var/log/httpd/
```

Sí tiene activa la opción de indice de directorio automático en el archivo de configuración del *Apache*; *IndexOptions* en el archivo *httpd.conf*, entonces podrá tener una parte de la seguridad a las peticiones que llegan a cada directorio. En mucho casos puede usted quiere que los usuarios accedan a la información desde un link, para ello cambiaremos los permisos de lectura sobre el directorio donde se encuentren los documentos, pero no a los documentos. Ejecutando los comandos:

```
[root@lhost ]/# cd /home/httpd/
[root@lhost ]/httpd# chmod 311 cutb
[root@lhost ]/httpd# ls -la
d-wx--x--x 13 webadmin webadmin 1024 Oct 19 08:12 cutb
```

Suponiendo que el directorio sea cutb.

Ahora, con esta modificación, las peticiones para este directorio serán regresadas con el mensaje de error :

```
Forbidden You don't have permission to access /cutb/ on this server.
```

3.5.2 Autenticando usuarios con el archivo .dbmpasswd. Este paso es necesario solamente si pensamos usar autenticación por contraseña para nuestro sitio Web. La autenticación por archivos de contraseñas es usada cuando tenemos la necesidad de proteger alguna parte de nuestro sitio web con una contraseña de usuario. Con el *Apache* existen muchas opciones de proteger nuestros nombres de usuarios y contraseñas.

El programa de utilidad del *Apache dbmmanager* puede usarse para crear y actualizar los usuarios y contraseñas de los usuarios *HTTP*.

Este método usa un formato *DBM* que es el mecanismo más rápido cuando tenemos más de mil usuarios. Primero que todo es importante cambiar los permisos de este programa a `750/-rwxr-x---`, permisos de escritura solo a los superusuarios *root*, escritura y ejecución al grupo y ningún permiso a los demás.

Para cambiar los permisos en el dbmmanager use el siguiente comando:

```
[root@lhost ]/# chmod 750 /usr/bin/dbmmanage
```

Para crear usuario y contraseña utilice este comando:

```
[root@lhost ]/# /usr/bin/dbmmanage /etc/httpd/.dbmpasswd
adduser username

New password:
Re-type new password:

User username added with password encrypted to l4jrdAL9MH0K.
```

Donde *</etc/httpd>* es la locación de el archivo de contraseña, *<.dbmpasswd>* es el nombre del archivo de contraseña y *<username>* es el nombre de usuario que queremos añadir en nuestro archivo de contraseña.

Sí usa el dbmmanager con su servidor Apache para crear las contraseñas y usuarios, no olvide incluir en su archivo de configuración */etc/httpd/conf/httpd.conf* la parte de su sitio que necesita con la autenticación de usuario por contraseña editando el archivo de configuración y añadiendo las líneas:

```
<Directory "/home/httpd/cutb/private">
Options None
AllowOverride AuthConfig
AuthName "restricted stuff"
AuthType Basic
AuthDBUserFile /etc/httpd/.dbmpasswd
require valid-user
</Directory>
```

El directorio *</home/httpd/cutb/private>* especifica el directorio que queremos proteger para los usuarios sin contraseñas. Para añadir el módulo de autenticación

por contraseña asegurese de instalar durante la configuración preinstalación del Apache el módulo *mod_auth_db.c* añadiendo las líneas:

```
--add-module=src/modules/standard/mod_auth_db.c
```

Luego de incluir los comandos y crear los usuarios reiniciamos nuestro servicio:

```
[root@lhost ]/# /etc/rc.d/init.d/httpd restart
Shutting down http: [ OK ]
Starting httpd: [ OK ]
```

Finalmente nosotros debemos probar nuestro directorio protegido accediendo desde un browser de la siguiente manera http://servidor_web.dominio/private/. El explorador pedira nombre de usuario y contraseña para el ingreso a este directorio.

3.5.3 Inmunizando nuestro archivo httpd.conf. Como nosotros sabemos es mejor prevenir que lamentar, cada vez que configuremos nuestro Apache inmunicemos su archivo de configuración para evitar que sea borrado, sobrescrito

ó crear un link simbólico sobre el archivo, para inmunizarlo basta con ejecutar el comando:

```
[root@lhost ]/# chattr +i /etc/httpd/conf/httpd.conf
```

3.5.4 Enjaulando a nuestro servidor Apache en el directorio chroot. En esta parte nos enfocaremos en la prevención del Apache desde su comienzo de uso como punto de quiebre en nuestro sistema de servidor. El Apache por defecto se ejecuta con un usuario *no root*, el cual limita ante cualquier daño ya que puede ser un usuario normal con su shell local.

Por supuesto, si el servidor permite la entrada de usuarios anónimos este mecanismo de seguridad es poco.

Por ende es mejor dar el paso y enjaularlo en el directorio *chroot*.

El beneficio principal de este directorio es que limita la porción de el sistema de archivo que nuestro demonio puede ver a el directorio raíz. Adicionalmente este directorio (ó partición) solamente soportará a nuestro servidor Apache, los programas incluidos se ejecutan extremadamente limitados, y lo más importante

no es necesario cambiar los *UID* a los programas del *root*, los cuales se pueden usar para ganar un acceso de *root*.

Este paso no es nada fácil y tiende a bloquear el servicio por alguna mala configuración. Antes de enjaular nuestro servicio debemos analizar las siguientes contras y pros para decidir lo mejor de nuestro sistema.

Pros

- Sí el *Apache* se ve comprometido, el atacante no podrá acceder a todo el sistema de archivos.

- Los scripts *CGI* que intenten acceder a nuestro servidor no se ejecutarán.

Contras

- Las librerías extras necesitamos tenerlas en el directorio *chroot* para que el Apache pueda trabajar.

- Si usamos algún Perl/CGI con el Apache, necesitamos copiar los binarios, las librerías y archivos apropiados en el directorio *chroot*. Aplica a los módulos de terceros y programas que interactúen con nuestro servidor Web.

La configuración supone que se instaló el Apache con el módulo *mod_ssl*. La diferencia radica en las librerías y binarios que debemos copiar en el directorio.

Recordemos que si compilamos el Apache para usar *mod_perl* debemos copiar los binarios y las librerias Perl, al directorio *chroot*. Los archivos del Perl residen en */usr/lib/perl5* y en caso de usar los recursos del Perl copie el directorio */chroot/httpd/usr/lib/perl5/*. No olvidemos crear antes el directorio */chroot/httpd/usr/lib/perl5* antes de copiar el directorio.

Los siguientes son pasos necesarios para ejecutar el Apache Web Server en el directorio *chroot*.

Nosotros debemos encontrar las librerías de *httpd*. Estas librerías debemos copiarlas en el directorio después. Para buscar las librerías ejecutamos:

```
[root@lhost ]/# ldd /usr/sbin/httpd
libpam.so.0 =>/lib/libpam.so.0 (0x40016000)
```

```
libm.so.6 =>/lib/libm.so.6 (0x4001f000)
libdl.so.2 =>/lib/libdl.so.2 (0x4003b000)
libcrypt.so.1 =>/lib/libcrypt.so.1 (0x4003e000)
libnsl.so.1 =>/lib/libnsl.so.1 (0x4006b000)
libresolv.so.2 =>/lib/libresolv.so.2 (0x40081000)
libdb.so.3 =>/lib/libdb.so.3 (0x40090000)
libc.so.6 =>/lib/libc.so.6 (0x400cb000)
/lib/ld-linux.so.2 =>/lib/ld-linux.so.2 (0x40000000)
```

Hagamos una nota de los archivos y continuamos.

Añadimos un nuevo *UID* y un nuevo *GID* para que estos ejecuten el *Apache httpd*.

Este paso es importante ya que ejecutarlo desde *root* perdemos los privilegios de enjaular el servicio y usando un diferente *UID* que ya exista puede acceder a otros recursos.

Consideremos que el servidor se ejecute como usuario1, un cracker puede acceder como usuario1 y ejecutar procesos dados al usuario1.

Presentamos un ejemplo de *UID/GID*. MIremos los archivos */etc/passwd* y */etc/group* para un número *UID/GID* libre. En nuestra configuración debemos usar el valor númerico de 80 y *UID/GID www*.

Los comandos anteriores crearán el grupo *www* con el valor numérico *GID* de 80, y el usuario *www* con valor numérico de *UID* 80.

Para configurar el *chroot*, primero necesitamos crear en él la estructura del Apache. Nosotros usamos */chroot/httpd* para nuestro Apache. Este es solamente un directorio en una partición distinta donde colocaremos al Apache para mayor seguridad.

```
[root@lhost ]/# /etc/rc.d/init.d/httpd stop
```

Solamente si el Apache está en servicio en nuestro sistema.

```
Shutting down http: [ OK ]
[root@lhost ]/# mkdir /chroot/httpd
```

Ahora creamos el resto de directorios:

```
[root@lhost ]/# mkdir /chroot/httpd/dev
[root@lhost ]/# mkdir /chroot/httpd/lib
[root@lhost ]/# mkdir /chroot/httpd/etc
[root@lhost ]/# mkdir -p /chroot/httpd/usr/sbin
[root@lhost ]/# mkdir -p /chroot/httpd/var/run
[root@lhost ]/# mkdir -p /chroot/httpd/var/log/httpd
[root@lhost ]/# chmod 750 /chroot/httpd/var/log/httpd/
[root@lhost ]/# mkdir -p /chroot/httpd/home/httpd
```

Copiamos los archivos principales, configuración, librerías y demás archivos para el funcionamiento de nuestro servidor:

```
[root@lhost ]/# cp -r /etc/httpd /chroot/httpd/etc/
[root@lhost ]/# cp -r /home/httpd/cgi-bin
/chroot/httpd/home/httpd/
[root@lhost ]/# cp -r /home/httpd/your-DocumentRoot
/chroot/httpd/home/httpd/
[root@lhost ]/# mknod /chroot/httpd/dev/null c 1 3
[root@lhost ]/# chmod 666 /chroot/httpd/dev/null
[root@lhost ]/# cp /usr/sbin/httpd /chroot/httpd/usr/sbin/
[root@lhost ]/# useradd -c "Apache Server" -u 80 -s
```

```
/bin/false -r -d /home/httpd www 2>/dev/null || :
```

Nosotros necesitaremos los siguientes directorios */chroot/httpd/etc*, */chroot/httpd/dev*, */chroot/httpd/lib*, */chroot/httpd/usr/sbin*, */chroot/httpd/var/run*, */chroot/httpd/home/httpd* y */chroot/httpd/var/log/httpd* ya que se supone que el chroot será nuestro directorio raíz para el Apache.

Sí compilamos el Apache para uso de *SSL*, debemos copiar el contenido de */etc/ssl* donde se almacena la clave pública y privada de nuestro servidor.

```
[root@lhost ]/# cp -r /etc/ssl /chroot/httpd/etc/
[root@lhost ]/# chmod 600 /chroot/httpd/etc/ssl/certs/ca.crt
[root@lhost ]/# chmod 600
/chroot/httpd//etc/ssl/certs/server.crt
[root@lhost ]/# chmod 600
/chroot/httpd/etc/ssl/private/ca.key
[root@lhost ]/# chmod 600
/chroot/httpd/etc/ssl/private/server.key
```

Desde que compilamos el Apache para usar ciertas librerias, entonces debemos instalar estas librerias en el directorio *chroot*. Usamos:

```
ldd /chroot/httpd/usr/sbin/httpd
```

Para buscar las librerias necesarias. La salida es dependiente a como hayamos compilado con el Apache puede ser algo similar a:

```
libpam.so.0 =>/lib/libpam.so.0 (0x40016000)
libm.so.6 =>/lib/libm.so.6 (0x4001f000)
libdl.so.2 =>/lib/libdl.so.2 (0x4003b000)
libcrypt.so.1 =>/lib/libcrypt.so.1 (0x4003e000)
libnsl.so.1 =>/lib/libnsl.so.1 (0x4006b000)
libresolv.so.2 =>/lib/libresolv.so.2 (0x40081000)
libdb.so.3 =>/lib/libdb.so.3 (0x40090000)
libc.so.6 =>/lib/libc.so.6 (0x400cb000)
/lib/ld-linux.so.2 =>/lib/ld-linux.so.2 (0x40000000)
```

Copiamos la libreria de esta manera:

```
[root@lhost ]/# cp /lib/libpam.so.0 /chroot/httpd/lib/
[root@lhost ]/# cp /lib/libm.so.6 /chroot/httpd/lib/
[root@lhost ]/# cp /lib/libdl.so.2 /chroot/httpd/lib/
```

```
[root@lhost ]/# cp /lib/libcrypt.so.1 /chroot/httpd/lib/
[root@lhost ]/# cp /lib/libnsl* /chroot/httpd/lib/
[root@lhost ]/# cp /lib/libresolv* /chroot/httpd/lib/
[root@lhost ]/# cp /lib/libdb.so.3 /chroot/httpd/lib/
[root@lhost ]/# cp /lib/libc.so.6 /chroot/httpd/lib/
[root@lhost ]/# cp /lib/ld-linux.so.2 /chroot/httpd/lib/
```

Es caso que necesitemos las siguientes extra librerías para algunas funciones de red, como resolver los nombres DNS. Así que las copiamos:

```
[root@lhost ]/# cp /lib/libnss_compat* /chroot/httpd/lib/
[root@lhost ]/# cp /lib/libnss_dns* /chroot/httpd/lib/
[root@lhost ]/# cp /lib/libnss_files* /chroot/httpd/lib/
```

Nosotros necesitamos copiar las contraseñas y los archivos de grupos dentro de */chroot/httpd/etc* . El concepto es parecido a los usuarios *ftpd* así como contraseñas y grupos. Ahora removemos todas las entradas excepto el usuario que ejecutará el Apache así como su contraseña y su grupo:

```
[root@lhost ]/# cp /etc/passwd /chroot/httpd/etc/
[root@lhost ]/# cp /etc/group /chroot/httpd/etc/
```

Editamos el archivo de contraseñas */chroot/httpd/etc/passwd* y borramos todo excepto el que ejecutará nuestro servidor *www*:

```
www:x:80:80::/home/www:/bin/bash
```

Editamos el archivo de grupo y hacemos lo mismo que el anterior dejando la entrada:

```
www:x:80:
```

Necesitamos ahora los archivos */etc/resolv.conf*, */etc/nsswitch.conf* y */etc/hosts* en nuestro directorio *chroot*:

```
[root@lhost ]/# cp /etc/resolv.conf /chroot/httpd/etc/
[root@lhost ]/# cp /etc/hosts /chroot/httpd/etc/
[root@lhost ]/# cp /etc/nsswitch.conf /chroot/httpd/etc/
```

Ahora debemos pasar inmutables algunos archivos para mayor seguridad, Pasamos a inmutables por archivo:

```
[root@lhost ]/# cd /chroot/httpd/etc/
[root@lhost ]/# chattr +i passwd
```

Pasamos a inmutable por grupo:

```
[root@lhost ]/# cd /chroot/httpd/etc/
[root@lhost ]/# chattr +i group
```

Pasamos a inmutable los archivos:

```
[root@lhost ]/# cd /chroot/httpd/etc/httpd/conf/
[root@lhost ]/# chattr +i httpd.conf
[root@lhost ]/# cd /chroot/httpd/etc/
[root@lhost ]/# chattr +i resolv.conf
[root@lhost ]/# cd /chroot/httpd/etc/
[root@lhost ]/# chattr +i hosts.113
[root@lhost ]/# cd /chroot/httpd/etc/
[root@lhost ]/# chattr +i nsswitch.conf
```

Copiampos el archivo de tiempo local para que las entradas del log sean ajustado a nuestra zona horaria apropiadamente:

```
[root@lhost ]/# cp /etc/localtime /chroot/httpd/etc/
```

Removemos los directorios ya no necesarios para el Apache:

```
[root@lhost ]/# rm -rf /var/log/httpd/
[root@lhost ]/# rm -rf /etc/httpd/
[root@lhost ]/# rm -rf /home/httpd/
[root@lhost ]/# rm -f /usr/sbin/httpd
```

Nosotros podemos remover todos los archivos y directorios ya que los contenemos en nuestro directorio *chroot*.

Normalmente los procesos llamados a syslogd desde */dev/log*, no son posibles *syslogd* necesita ser llamado a */chroot/httpd/dev/log*.

Para hacer esto editamos el archivo syslog */etc/rc.d/init.d/syslog* y especificamos los sitios adicionales a escuchar *daemon syslogd -m 0* para que lea:

```
daemon syslogd -m 0 -a /chroot/httpd/dev/log
```

Por defecto el script *httpd* comienza el demonio *httpd* fuera del *chroot*. Nosotros debemos cambiar esto para que inicie desde el *chroot*.

Para esto editamos el script *httpd /etc/rc.d/init.d/httpd* y cambiamos la línea `daemon httpd a`:

```
/usr/sbin/chroot /chroot/httpd/ /usr/sbin/httpd -DSSL
```

y:

```
rm -f /var/run/httpd.pid
```

a:

```
rm -f /chroot/httpd/var/run/httpd.pid
```

Finalmente debemos probar la nueva configuración chroot de nuestro servidor web Apache. Primero debemos reiniciar nuestro demonio *syslogd* con el siguiente comando:

```
[root@lhost ]/# /etc/rc.d/init.d/syslog restart
```

```
Shutting down kernel logger: [ OK ]
Shutting down system logger: [ OK ]
Starting system logger: [ OK ]
Starting kernel logger: [ OK ]
```

Ahora iniciamos el servidor Apache con :

```
[root@lhost ]/# /etc/rc.d/init.d/httpd start
Starting httpd: [ OK ]
```

Si no hay errores podemos hacer un llamado ps ó grep y vemos si se está ejecutando:

```
[root@lhost ]/# ps ax | grep httpd
14373 ? S 0:00 httpd -DSSL
14376 ? S 0:00 httpd -DSSL
14377 ? S 0:00 httpd -DSSL
14378 ? S 0:00 httpd -DSSL
14379 ? S 0:00 httpd -DSSL
14380 ? S 0:00 httpd -DSSL
14381 ? S 0:00 httpd -DSSL
```

```
14382 ? S 0:00 httpd -DSSL
14383 ? S 0:00 httpd -DSSL
14384 ? S 0:00 httpd -DSSL
14385 ? S 0:00 httpd -DSSL
14386 ? S 0:00 httpd -DSSL
14387 ? S 0:00 httpd -DSSL
14388 ? S 0:00 httpd -DSSL
14389 ? S 0:00 httpd -DSSL
14390 ? S 0:00 httpd -DSSL
14391 ? S 0:00 httpd -DSSL
14397 ? S 0:00 httpd -DSSL
14476 ? S 0:00 httpd -DSSL
14477 ? S 0:00 httpd -DSSL
14478 ? S 0:00 httpd -DSSL
```

Ahora asegurémonos que chroot esté fuera de los número de procesos ejecutando:

```
[root@lhost ]/# ls -la /proc/14373/root/
```

Si vemos :

```
dev
etc
home
lib
usr
var
```

Esto quiere decir que se hizo lo correcto.

Volvemos a mencionar si usamos Perl necesitamos copiar los archivos binarios y librerias perl */usr/lib/perl5*, en el área chroot. Lo mismo aplica para *SSL*, *PHP*, *LDAP*, *PostgresSQL* y otros programas.

Ahora los archivos de log del Apache residentes en el directorio */chroot/var/log/httpd* ya no estan en el directorio */var/log/httpd* y por esta razón nosotros necesitamos modificar el archivo */etc/logrotate.d/httpd*. Si habíamos compilado Apahe con *mod_ssl* debemos añadir una línea mas para permitir que el programa *logrotate* rote archivos de *log* de *ssl*.

Configuramos nuestro archivo */etc/logrotate.d/apache* para rotar los archivos de *log* cada semana automáticamente.

Creamos el archivo apache, con el comando `touch /etc/logrotate.d/apache` y añadimos:

```
/chroot/httpd/var/log/httpd/access_log {
missingok
postrotate
/usr/bin/killall -HUP /chroot/httpd/usr/sbin/httpd
endscript
} /chroot/httpd/var/log/httpd/error_log {
missingok
postrotate
/usr/bin/killall -HUP /chroot/httpd/usr/sbin/httpd
endscript
} /chroot/httpd/var/log/httpd/ssl_request_log {
missingok
postrotate
/usr/bin/killall -HUP /chroot/httpd/usr/sbin/httpd
endscript
```

```
} /chroot/httpd/var/log/httpd/ssl_engine_log {
missingok
postrotate.116
/usr/bin/killall -HUP /chroot/httpd/usr/sbin/httpd
endscript
}
```

3.6 Optimización del Apache.

3.6.1 El módulo mod_mmap_static del Apache. Este es un módulo especial que puede ser usado para aumentar el desempeño de nuestro servidor. Este módulo trabaja dando un mapeado de estadísticas configurados en una lista de peticiones frecuentes, pero no cambia los archivos en su directorio raíz. Si los archivos que nuestro servidor maneja no son cambiados constantemente podemos usar este módulo para que coloque estos documentos estáticos en memoria e incremente la velocidad de nuestro servidor.

Es importante notar que este módulo debe estar activo desde la configuración e instalación de nuestro servidor para que pueda ser usado. Para esto debemos

incluir la línea a nuestro archivo de configuración `--add-module-../mod_mmap_static.c`.

Los documentos que irán a memoria deben incluirse con el siguiente comando:

```
[root@lhost ]/# find /home/httpd/cutb -type f -print | sed –e 's/.*/mmapfile &/' /etc/httpd/conf/mmap.conf
```

Donde `/home/httpd/cutb` es el directorio raíz, ó para ser más preciso el directorio que será servido los documentos y `/etc/httpd/conf/mmap.conf` es la locación donde nosotros queremos crear este archivo, que contendrá el mapa de los documentos en nuestro directorio raíz.

Cada vez que el archivo *mmap.conf* sea creado debemos incluirlo en el archivo de configuración del Apache *httpd.conf* para que estas opciones sean activas en nuestro servidor.

Así editamos el archivo */etc/httpd/conf/httpd.conf* y añadimos la línea:

```
<IfModule mod_include.c>
```

```
Include conf/mmap.conf
</IfModule>
```

Recuerde que esta opción solo con los documentos que no cambien constantemente en el sitio, de lo contrario puede provocar un error de acceso perdido.

Debemos reiniciar el Apache para que los cambios surjan efectos:

```
[root@lhost ]/# /etc/rc.d/init.d/httpd restart
Shutting down http: [ OK ]
Starting httpd: [ OK ]
```

3.6.2 Los atributos atime y noatime. Los atributos atime y noatime de Linux pueden ser usados para obtener un gran desempeño en el Apache. Linux tiene un montaje especial para los filesystem llamado noatime que puede ser adherido a cada línea de la dirección de un archivo en */etc/fstab*. Si el sistema de archivos fue montado con esta opción, leer el acceso desde el sistema de archivos . La importancia de esta configuración es que elimina la necesidad del sistema de escribir al sistema de archivos para archivos que solamente van a ser leídos.

Ya que el proceso de escritura es más complicado y esto hace que el desempeño aumente. Hay que notar que podemos escribir información en un archivo y actualizarlo y luego volver a escribir.

A continuación activaremos la opción *noatime* a nuestro directorio *chroot*. Como primera medida editamos el archivo */etc/fstab* y añadimos en la línea referida a *chroot* la opción *noatime* después de las opciones por defecto como aparece a continuación:

```
/dev/sda7 /chroot ext2 defaults,noatime 1 2
```

Nosotros no necesitamos reiniciar el sistema, simplemente debemos remontar la partición que usamos de la siguiente manera:

```
[root@lhost] /#mount -oremount /chroot/
```

Podemos probar nuestro resultados con el comando siguiente:

```
[root@lhost]# cat /proc/mounts
/dev/root / ext2 rw 0 0
```

```
/proc /proc proc rw 0 0
/dev/sda1 /boot ext2 rw 0 0
/dev/sda8 /cache ext2 rw 0 0
/dev/sda7 /chroot ext2 rw,noatime 0 0
/dev/sda6 /home ext2 rw 0 0
/dev/sda11 /tmp ext2 rw 0 0
/dev/sda5 /usr ext2 rw 0 0
/dev/sda9 /var ext2 rw 0 0
none /dev/pts devpts rw 0 0
```

Podemos ver que nuestra partición se encuentra con la opción activa.

3.7 ARCHIVOS INSTALADOS.

3.7.1 Archivos instalados por la biblioteca de funciones MM. Los archivos instalados por la librería MM son:

```
/usr/bin/mm-config
/usr/include/mm.h
/usr/lib/libmm.la
```

```
/usr/lib/libmm.a
/usr/man/man1/mm-config.1
/usr/man/man3/mm.3
```

3.7.2 Archivos instalados por el Apache Web Server. Los archivos instalados por el servidor web Apache son:

```
/etc/rc.d/init.d/httpd /etc/rc.d/rc0.d/K15httpd
/etc/rc.d/rc2.d/K15httpd /etc/rc.d/rc3.d/S85httpd
/etc/rc.d/rc5.d/S85httpd /etc/rc.d/rc6.d/K15httpd
/etc/httpd /etc/httpd/conf
/etc/httpd/conf/httpd.conf /etc/httpd/conf/mime.types.default
/etc/httpd/conf/magic.default /etc/httpd/conf/magic
/home/httpd /home/httpd/cgi-bin
/home/httpd/cgi-bin/test-cgi /usr/bin/htpasswd
/usr/bin/dbmmanage /usr/include/apache
/usr/include/apache/xml/asciitab.h
/usr/include/apache/xml/hashtable.h
/usr/include/apache/xml/latin1tab.h
/usr/include/apache/xml/nametab.h
```

```
/usr/include/apache/xml/xmldef.h
/usr/include/apache/xml/xmlparse.h
/usr/include/apache/xml/xmltok.h
/usr/include/apache/xml/xmltok_impl.h
/usr/include/apache/ap.h /usr/include/apache/ap_compat.h
/usr/include/apache/ap_config_auto.h
/usr/include/apache/ap_ctx.h
/usr/include/apache/ap_hook.h /usr/include/apache/ap_md5.h
/usr/include/apache/ap_mmn.h /usr/include/apache/ap_sha1.h
/usr/include/apache/compat.h /usr/include/apache/conf.h
/usr/include/apache/fnmatch.h /usr/include/apache/hsregex.h
/usr/include/apache/http_config.h
/usr/include/apache/http_core.h
/usr/include/apache/http_main.h
/usr/include/apache/http_protocol.h
/usr/include/apache/http_vhost.h /usr/include/apache/httpd.h
/usr/include/apache/rfc1413.h
/usr/include/apache/scoreboard.h
/usr/include/apache/util_md5.h
/usr/include/apache/util_script.h
/usr/include/apache/os.h /usr/include/apache/os-inline.c
```

```
/usr/man/man1/htpasswd.1 /usr/man/man1/htdigest.1
/usr/man/man8/ab.8 /usr/man/man8/httpd.8.119
/usr/man/man8/rotatelogs.8 /usr/man/man8/apxs.8
/usr/sbin/ab /usr/sbin/logresolve
/usr/sbin/apxs /var/log/httpd
/var/cache/httpd
/etc/rc.d/rc1.d/K15httpd
/etc/rc.d/rc4.d/S85httpd
/etc/logrotate.d/apache
/etc/httpd/conf/httpd.conf.default
/etc/httpd/conf/mime.types
/etc/httpd/php.ini
/home/httpd/cgi-bin/printenv
/usr/bin/htdigest
/usr/include/apache/xml
/usr/include/apache/xml/iasciitab.h
/usr/include/apache/xml/utf8tab.h
/usr/include/apache/xml/xmlrole.h
/usr/include/apache/alloc.h
/usr/include/apache/ap_config.h
/usr/include/apache/ap_ctype.h
```

```
/usr/include/apache/ap_mm.h
/usr/include/apache/buff.h
/usr/include/apache/explain.h
/usr/include/apache/http_conf_globals.h
/usr/include/apache/http_log.h
/usr/include/apache/http_request.h
/usr/include/apache/multithread.h
/usr/include/apache/util_date.h
/usr/include/apache/util_uri.h
/usr/lib/apache
/usr/man/man1/dbmmanage.1
/usr/man/man8/logresolve.8
/usr/sbin/httpd
/usr/sbin/rotatelogs
/var/cache.120
```

3.7.3 Archivos instalados por el Server Side PHP4, con el Apache Web Server.

Los archivos instalados por el módulo server side PHP4 son:

```
/usr/bin/phpize /usr/bin/php-config
```

```
/usr/include/php /usr/include/php/Zend
/usr/include/php/Zend/FlexLexer.h
/usr/include/php/Zend/acconfig.h
/usr/include/php/Zend/modules.h
/usr/include/php/Zend/zend-parser.h
/usr/include/php/Zend/zend-scanner.h
/usr/include/php/Zend/zend.h
/usr/include/php/Zend/zend_API.h
/usr/include/php/Zend/zend_alloc.h
/usr/include/php/Zend/zend_builtin_functions.h
/usr/include/php/Zend/zend_compile.h
/usr/include/php/Zend/zend_config.h
/usr/include/php/Zend/zend_config.w32.h
/usr/include/php/Zend/zend_constants.h
/usr/include/php/Zend/zend_dynamic_array.h
/usr/include/php/Zend/zend_errors.h
/usr/include/php/Zend/zend_execute.h
/usr/include/php/Zend/zend_execute_locks.h
/usr/include/php/Zend/zend_extensions.h
/usr/include/php/Zend/zend_fast_cache.h
/usr/include/php/Zend/zend_globals.h
```

```
/usr/include/php/Zend/zend_globals_macros.h
/usr/include/php/Zend/zend_hash.h
/usr/include/php/Zend/zend_highlight.h
/usr/include/php/Zend/zend_indent.h
/usr/include/php/Zend/zend_list.h
/usr/include/php/Zend/zend_llist.h
/usr/include/php/Zend/zend_operators.h
/usr/include/php/Zend/zend_ptr_stack.h
/usr/include/php/Zend/zend_stack.h
/usr/include/php/Zend/zend_variables.h
/usr/include/php/TSRM /usr/include/php/TSRM/TSRM.h
/usr/include/php/ext /usr/include/php/ext/standard
/usr/include/php/ext/standard/base64.h
/usr/include/php/ext/standard/basic_functions.h
/usr/include/php/ext/standard/cyr_convert.h
/usr/include/php/ext/standard/datetime.h
/usr/include/php/ext/standard/dl.h
/usr/include/php/ext/standard/dns.h
/usr/include/php/ext/standard/exec.h
/usr/include/php/ext/standard/file.h
```

```
/usr/include/php/ext/standard/flock_compat.h
/usr/include/php/ext/standard/fsock.h
/usr/include/php/ext/standard/global.h
/usr/include/php/ext/standard/head.h
/usr/include/php/ext/standard/html.h
/usr/include/php/ext/standard/info.h
/usr/include/php/ext/standard/md5.h
/usr/include/php/ext/standard/microtime.h
/usr/include/php/ext/standard/pack.h
/usr/include/php/ext/standard/pageinfo.h
/usr/include/php/ext/standard/php_array.h
/usr/include/php/ext/standard/php_assert.h
/usr/include/php/ext/standard/php_browscap.h
/usr/include/php/ext/standard/php_crypt.h
/usr/include/php/ext/standard/php_dir.h
/usr/include/php/ext/standard/php_filestat.h
/usr/include/php/ext/standard/php_image.h
/usr/include/php/ext/standard/php_iptc.h
/usr/include/php/ext/standard/php_lcg.h
/usr/include/php/ext/standard/php_link.h
```

```
/usr/include/php/ext/standard/php_mail.h
/usr/include/php/ext/standard/php_metaphone.h
/usr/include/php/ext/standard/php_output.h
/usr/include/php/ext/standard/php_rand.h
/usr/include/php/ext/standard/php_standard.h
/usr/include/php/ext/standard/php_string.h
/usr/include/php/ext/standard/php_syslog.h
/usr/include/php/ext/standard/php_var.h
/usr/include/php/ext/standard/phpdir.h
/usr/include/php/ext/standard/phpmath.h.121
/usr/include/php/ext/standard/quot_print.h
/usr/include/php/ext/standard/reg.h
/usr/include/php/ext/standard/type.h
/usr/include/php/ext/standard/uniqid.h
/usr/include/php/ext/standard/url.h
/usr/include/php/ext/standard/url_scanner.h
/usr/include/php/regex
/usr/include/php/regex/regex.h
/usr/include/php/regex/regex_extra.h
/usr/include/php/php.h
/usr/include/php/php_regex.h
```

```
/usr/include/php/php3_compat.h
/usr/include/php/safe_mode.h
/usr/include/php/fopen-wrappers.h
/usr/include/php/php_version.h
/usr/include/php/php_globals.h
/usr/include/php/php_reentrancy.h
/usr/include/php/php_ini.h
/usr/include/php/SAPI.h
/usr/include/php/php_config.h
/usr/include/php/zend_config.h
/usr/include/php/build-defs.h
/usr/lib/php /usr/lib/php/DB
/usr/lib/php/DB/common.php
/usr/lib/php/DB/odbc.php
/usr/lib/php/DB/mysql.php
/usr/lib/php/DB/pgsql.php
/usr/lib/php/DB/storage.php
/usr/lib/php/build
/usr/lib/php/build/pear.m4
/usr/lib/php/build/fastgen.sh
/usr/lib/php/build/library.mk
```

```
/usr/lib/php/build/ltlib.mk
/usr/lib/php/build/program.mk
/usr/lib/php/build/rules.mk
/usr/lib/php/build/rules_pear.mk
/usr/lib/php/build/shtool
/usr/lib/php/build/acinclude.m4
/usr/lib/php/DB.php
```

3.7.4 Archivos instalados por el mod_perl. Los archivos instalados por el módulo de Perl son:

```
/usr/lib/perl5/5.00503/i386-linux/perllocal.pod
/usr/lib/perl5/man/man3/Apache.3
/usr/lib/perl5/man/man3/Apache::Constants.3
/usr/lib/perl5/man/man3/Apache::Leak.3
/usr/lib/perl5/man/man3/Apache::Log.3
/usr/lib/perl5/man/man3/Apache::PerlRunXS.3
/usr/lib/perl5/man/man3/Apache::Symbol.3
/usr/lib/perl5/man/man3/Apache::Table.3
/usr/lib/perl5/man/man3/Apache::URI.3
```

```
/usr/lib/perl5/man/man3/Apache::Util.3
/usr/lib/perl5/man/man3/Apache::FakeRequest.3
/usr/lib/perl5/man/man3/mod_perl.3
/usr/lib/perl5/man/man3/Apache::ExtUtils.3
/usr/lib/perl5/man/man3/Apache::SIG.3
/usr/lib/perl5/man/man3/Apache::Status.3
/usr/lib/perl5/man/man3/Apache::Include.3
/usr/lib/perl5/man/man3/Apache::Debug.3
/usr/lib/perl5/man/man3/Apache::Resource.3
/usr/lib/perl5/man/man3/Apache::src.3.122
/usr/lib/perl5/man/man3/Apache::PerlRun.3
/usr/lib/perl5/man/man3/Apache::httpd_conf.3
/usr/lib/perl5/man/man3/mod_perl_traps.3
/usr/lib/perl5/man/man3/Apache::Options.3
/usr/lib/perl5/man/man3/mod_perl_cvs.3
/usr/lib/perl5/man/man3/Apache::Symdump.3
/usr/lib/perl5/man/man3/Apache::RegistryLoader.3
/usr/lib/perl5/man/man3/mod_perl_method_handlers.3
/usr/lib/perl5/man/man3/mod_perl_tuning.3
/usr/lib/perl5/man/man3/cgi_to_mod_perl.3
/usr/lib/perl5/man/man3/Apache::StatINC.3
```

```
/usr/lib/perl5/man/man3/Apache::Registry.3
/usr/lib/perl5/man/man3/Bundle::Apache.3
/usr/lib/perl5/man/man3/Apache::SizeLimit.3
/usr/lib/perl5/man/man3/Apache::PerlSections.3
/usr/lib/perl5/man/man3/Apache::RedirectLogFix.3
/usr/lib/perl5/site_perl/5.005/i386-linux/auto
/usr/lib/perl5/site_perl/5.005/i386-linux/auto/Apache
/usr/lib/perl5/site_perl/5.005/i386-linux/auto/Apache/include
/usr/lib/perl5/site_perl/5.005/i386-
Linux/auto/Apache/include/include
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/buff.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/multithread.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/httpd.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/ap_config.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/alloc.h
```

```
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/ap.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/ap_md5.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/ap_ctx.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/util_md5.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/rfc1413.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/conf.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/util_uri.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/explain.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/ap_compat.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/http_config.h
```

```
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/ap_sha1.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/scoreboard.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/compat.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/http_request.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/http_core.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/ap_mm.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/http_protocol.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/util_date.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/ap_hook.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/http_main.h
/usr/lib/perl5/site_perl/5.005/i386-
```

```
linux/auto/Apache/include/include/http_conf_globals.h.123
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/util_script.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/http_vhost.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/ap_ctype.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/hsregex.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/ap_mmn.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/ap_config_auto.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/http_log.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/include/fnmatch.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/netware
```

```
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/netware/os.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/netware/getopt.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/netware/test_char.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/netware/uri_delims.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/netware/precomp.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/bs2000
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/bs2000/os-inline.c
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/bs2000/ebcdic.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/bs2000/os.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/tpf
```

```
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/tpf/ebcdic.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/tpf/os.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/tpf/os-inline.c
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/win32
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/win32/service.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/win32/getopt.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/win32/registry.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/win32/resource.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/win32/installer
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/win32/installer/installdll
/usr/lib/perl5/site_perl/5.005/i386-
```

linux/auto/Apache/include/os/win32/installer/installdll/test

/usr/lib/perl5/site_perl/5.005/i386-

linux/auto/Apache/include/os/win32/installer/installdll/test/

test.h

/usr/lib/perl5/site_perl/5.005/i386-

linux/auto/Apache/include/os/win32/installer/installdll/test/

resource.h

/usr/lib/perl5/site_perl/5.005/i386-

linux/auto/Apache/include/os/win32/os.h

/usr/lib/perl5/site_perl/5.005/i386-

linux/auto/Apache/include/os/win32/passwd.h

/usr/lib/perl5/site_perl/5.005/i386-

linux/auto/Apache/include/os/win32/readdir.h

/usr/lib/perl5/site_perl/5.005/i386-

linux/auto/Apache/include/os/unix

/usr/lib/perl5/site_perl/5.005/i386-

linux/auto/Apache/include/os/unix/os.h

/usr/lib/perl5/site_perl/5.005/i386-

linux/auto/Apache/include/os/unix/os-inline.c

/usr/lib/perl5/site_perl/5.005/i386-

linux/auto/Apache/include/os/os390

```
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/os390/os-inline.c.124
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/os390/ebcdic.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/os390/os.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/mpeix
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/mpeix/os-inline.c
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/mpeix/os.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/os2
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/os2/os.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/os/os2/os-inline.c
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules
```

```
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/ssl
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/ssl/ssl_expr.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/ssl/ssl_util_table.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/ssl/ssl_util_ssl.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/ssl/ssl_expr_parse.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/ssl/mod_ssl.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/ssl/ssl_util_sdbm.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/perl
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/perl/mod_perl.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/perl/mod_perl_version.h
```

```
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/perl/perl_PL.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/perl/mod_perl_xs.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/php4
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/php4/mod_php4.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/proxy
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/proxy/mod_proxy.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/standard
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/modules/standard/mod_rewrite.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/support
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/support/suexec.h
```

```
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/lib
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/lib/expat-lite
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/lib/expat-lite/iasciitab.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/lib/expat-lite/latin1tab.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/lib/expat-lite/xmldef.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/lib/expat-lite/
xmlparse.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/lib/expat-lite/xmltok.h.125
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/lib/expat-lite/xmlrole.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/lib/expat-lite/
hashtable.h
```

```
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/lib/expat-lite/nametab.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/lib/expat-lite/
xmltok_impl.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/lib/expat-lite/utf8tab.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/lib/expat-lite/asciitab.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/regex
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/regex/utils.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/regex/regex2.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/regex/cclass.h
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/include/regex/cname.h
/usr/lib/perl5/site_perl/5.005/i386-linux/auto/Apache/typemap
/usr/lib/perl5/site_perl/5.005/i386-linux/auto/Apache/Leak
```

```
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/Leak/Leak.so
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/Leak/Leak.bs
/usr/lib/perl5/site_perl/5.005/i386-linux/auto/Apache/Symbol
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/Symbol/Symbol.so
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Apache/Symbol/Symbol.bs
/usr/lib/perl5/site_perl/5.005/i386-linux/auto/mod_perl
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/mod_perl/.packlist
/usr/lib/perl5/site_perl/5.005/i386-linux/mod_perl.pod
/usr/lib/perl5/site_perl/5.005/i386-linux/Bundle
/usr/lib/perl5/site_perl/5.005/i386-linux/Bundle/Apache.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/test.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/Debug.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/Resource.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/src.pm
```

/usr/lib/perl5/site_perl/5.005/i386-
linux/Apache/httpd_conf.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/Symdump.pm
/usr/lib/perl5/site_perl/5.005/i386-
linux/Apache/RegistryLoader.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/Registry.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/SizeLimit.pm
/usr/lib/perl5/site_perl/5.005/i386-
linux/Apache/RedirectLogFix.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/MyConfig.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/Constants
/usr/lib/perl5/site_perl/5.005/i386-
linux/Apache/Constants/Exports.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/SIG.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/StatINC.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/Opcode.pm
/usr/lib/perl5/site_perl/5.005/i386-
linux/Apache/PerlSections.pm
/usr/lib/perl5/site_perl/5.005/i386-
linux/Apache/FakeRequest.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/ExtUtils.pm

```
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/Include.pm
/usr/lib/perl5/site_perl/5.005/i386-
linux/Apache/Status.pm.126
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/PerlRun.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/Options.pm
/usr/lib/perl5/site_perl/5.005/i386-
linux/Apache/RegistryNG.pm
/usr/lib/perl5/site_perl/5.005/i386-
linux/Apache/RegistryBB.pm
/usr/lib/perl5/site_perl/5.005/i386-
linux/Apache/Connection.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/Constants.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/File.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/Leak.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/Log.pm
/usr/lib/perl5/site_perl/5.005/i386-
linux/Apache/ModuleConfig.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/PerlRunXS.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/Server.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/Symbol.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/Table.pm
```

```
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/URI.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/Apache/Util.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/mod_perl_hooks.pm
/usr/lib/perl5/site_perl/5.005/i386-
linux/mod_perl_hooks.pm.PL
/usr/lib/perl5/site_perl/5.005/i386-linux/mod_perl_tuning.pod
/usr/lib/perl5/site_perl/5.005/i386-linux/mod_perl_cvs.pod
/usr/lib/perl5/site_perl/5.005/i386-
linux/mod_perl_method_handlers.pod
/usr/lib/perl5/site_perl/5.005/i386-linux/mod_perl.pm
/usr/lib/perl5/site_perl/5.005/i386-linux/mod_perl_traps.pod
/usr/lib/perl5/site_perl/5.005/i386-linux/cgi_to_mod_perl.pod
```

3.7.5 Archivos instalados por el módulo Devel::SysDump. Los archivos instalados por el módulo Devel::Symdump son:

```
/usr/lib/perl5/man/man3/Devel::Symdump.3
/usr/lib/perl5/site_perl/5.005/i386-linux/auto/Devel
/usr/lib/perl5/site_perl/5.005/i386-linux/auto/Devel/Symdump
```

```
/usr/lib/perl5/site_perl/5.005/i386-
linux/auto/Devel/Symdump/.packlist
/usr/lib/perl5/site_perl/5.005/Devel
/usr/lib/perl5/site_perl/5.005/Devel/Symdump
/usr/lib/perl5/site_perl/5.005/Devel/Symdump/Export.pm
/usr/lib/perl5/site_perl/5.005/Devel/Symdump.pm
```

3.7.6 Archivos instalados por la libreria CGI.pm para Perl. Los archivos instalados por la librería CGI.pm son:

```
/usr/lib/perl5/5.00503/CGI/Pretty.pm
/usr/lib/perl5/5.00503/i386-linux/auto/CGI
/usr/lib/perl5/5.00503/i386-linux/auto/CGI/.packlist
/usr/lib/perl5/man/man3/CGI::Pretty.3.127
```

4. EL SERVICIO FTP

El servicio de transferencia de archivos es actualmente poco usado debido a los problemas con respecto a la seguridad que este proporciona, ó prefieren usar otros protocolos para este fin. Sin embargo muchas empresas aún siguen usándolo, usado mucho también en redes internas para colocar ficheros de instalación para muchos de los programas.

Para un compedio de seguridad completa en este protocolo se hace necesario la creación de un programa especial para subir archivos y uno especial para bajarlos, de modo que los usuarios no puedan hacer peticiones diferentes a la de los archivos a servir he aquí el caso de servidores como filepool que permitían bajar la lista de archivos a servir para luego bajarla con un programa especial, permitiendo así la seguridad en este protocolo.

4.1 DISEÑO DEL SERVICIO FTP.

El diseño de este aunque al parecer puede ser muy sencillo a la hora de implementarlo puede ser muy tedioso. La principal parte del diseño se centra en

los permisos a ficheros y directorios que se quieran compartir. En materia de seguridad como hemos dicho es muy bajo sin importar el tipo de servidor *FTP* que se esté usando, así que sí en el diseño tenemos que por lo menos colocar los mínimos requerimientos de seguridad.

El mantenimiento de este servicio puede ser llevado a cabo por el administrador de la red, ó por el administrador del servicio *HTTP*, el mantenimiento no es muy continuo debido a que los ficheros y directorios a compartir no se actualizan diariamente.

Nosotros podemos configurar un *FTP* de dos formas, la primera como *FTP* privado donde solo usuarios autorizados tienen derecho a la información compartida; la segunda es como *FTP* anónimo donde cualquiera puede acceder a un servidor poniendo el nombre de usuario "*anonymous*" y luego colocando su dirección de e-mail válida.

Esta a pericia del administrador controlar las entradas y/ó llevar un registro de estas. Los modos de un *FTP* pueden ser distintos de acuerdo al uso que este tenga así pues este podrá transmitir en modo *binario*, *ASCII*, etc.

A continuación presentaremos instalación y configuración de uno de los servidores *FTP* más usados en la red, aunque no sea el más seguro. Generalmente todo este tipo de servidores posee *bugs* y huecos de seguridad, a medida del tiempo que se van solucionando van saliendo otros nuevos.

4.2 INSTALACION DEL SERVIDOR FTP WU-FTP

La instalación asume lo siguiente:

- Compatibilidad con comandos UNIX.
- El directorio fuente es /var/tmp.
- Todos los pasos de la instalación deben ser bajo el superusuario root.
- La versión de wu-ftp es la 2.6.0

Los paquetes puede descargarlos de la página http://www.wu-ftpd.org/.

Para comenzar necesitamos descomprimir el archivo tar que descargamos de la siguiente forma:

```
[root@lhost ] /# cp wu-ftpd-version.tar.gz /var/tmp
[root@lhost ] /# cd /var/tmp
[root@lhost ]/tmp# tar xzpf wu-ftpd-version.tar.gz
```

Como segundo paso debemos editar la configuración del archivo *../src/pathnames.h* con los argumentos como vemos en la tabla siguiente:

Argumento	**Descripción**
`_PATH_FTPUSERS /etc/ftpusers"`	El archivo que lista los usuarios que no pueden hacer una conexión *FTP*. Usualmente contiene *root* y todos los usuarios que no son conectados como una persona real (*bin*, *sync*, *nobody*, etc.).
`_PATH_FTPACCESS`	El archivo de configuración para el

/usr/local/etc/ftpaccess"	sistema.
_PATH_FTPHOSTS "/etc/ftphosts"	El archivo de configuración para cada usuario.
_PATH_EXECPATH "/bin/ftp-exec"	El directorio que contiene binarios adicionales para usar con el comando *SITE EXEC*. Este path es ralativo a la instalación del ftp. Para añadir funcionalidad igual para usuarios reales, creamos un link simbólico desde el sitio del ftp al nuevo directorio.
_PATH_PIDNAMES /usr/local/daemon/ftpd/ftp.pids- % s"	El nombre del archivo cambia para archivos pid. El *%s* adquiere el nombre remplazandose por el nombre de la clase correspondiente.

	Existen números de archivos *pid* tanto como clases en nuestro *ftpaccess*.
`_PATH_CVT "/usr/local/etc/ftpconversions"`	El archivo de configuración que contiene la conversión (file -> file.Z, etc).
`_PATH_XFERLOG "/usr/adm/xferlog"`	Es el archivo donde toda la información del log es almacenada.
`_PATH_PRIVATE "/etc/ftpgroups"`	Es el archivo para guardar las contraseñas de grupo, para ser usado con los comandos *SITE GROUP* y *SITE GRAS*.
`_PATH_UTMP "/etc/utmp"`	Es el lugar donde el archivo *utmp* esta localizado.
`_PATH_WTMP "/usr/adm/wtmp"`	Es el lugar donde el archivo *wtmp* esta localizado.
`_PATH_LASTLOG "/usr/adm/lastlog"`	Es el archivo del último log.
`_PATH_BSHELL "/bin/sh"`	Donde el Bourne Shell está localizado.

Tabla 6. Argumentos para la configuración del archivo *pathnames.h*

El tercer paso consiste en crear nuestro instalador así escribimos desde el *shell* *./build xxx*.

Podemos también especificar un compilador alterno de C por la entrada *./build CC=yyy xxx*. Donde *yyy* es el comando del compilador para remplazar al *cc* por ejemplo *./build CC=gcc xxx; y xxx* puede ser uno de los siguientes comandos:

- gen : hace una instalación genérica.
- aix : IBM AIX
- aux : AU/X
- bdi : BSD/OS
- bsd : BSD
- dec : DEC Unix 3.X
- du4 : DEC Unix 4.X or later

- dyn : Dynix
- fbs : FreeBSD 2.0 or later.130
- hiu : Hitachi Unix
- hpx : HP-UX
- lnx : Linux (tested on 2.0.30)
- nbs : NetBSD 1.X
- nx2 : NeXTstep 2.x
- nx3 : NeXTstep 3.x
- osf : OSF/1
- ptx: No se encontró información al respecto de su propia casa matriz.

- sco : SCO Unix 3.2v4.2/SCO OpenServer 5

- sgi : SGI Irix 4.0.5a

- sny : Sony NewsOS

- sol : SunOS 5.x / Solaris 2.x

- s41 : SunOS 4.1.x

- ult : Ultrix 4.x

- uxw : UnixWare 1.1 or later

- clean : Limpia los archivos de objeto y reduce el espacio ocupado en disco después de la instalación.

- install: Instala el ftpd.

Luego es importante darle al *FTP* una cuenta de shell que no sea real, para evitar intrusiones al sistema. Para crear el nuevo usuario FTP con mínimos privilegios podemos usar los comandos:

```
[root@lhost ] /# mkdir /home/ftp
[root@lhost ] /# useradd -d /home/ftp/ftpadmin/ -s /dev/null
ftpadmin > /dev/null 2>&1
[root@lhost ] /# passwd ftpadmin
Changing password for user ftpadmin
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

Ahora damos ejecutamos el comando `./buid install` desde el directorio de creación de instaladores y el servicio *FTP* se instalará, terminado esto borramos los archivos temporales usados en la instalación del directorio:

```
rm –r /var/tmp/nombre_directorio_wu-ftp_install.
```

Editamos el fichero */etc/inetd.conf* para referirnos al nuevo *ftpd*. En muchos casos, este paso no es necesario, como el paso de instalación envía el nuevo software sobre el antiguo.

Si intenta actualizar usando extensiones de este servidor debe asegurarse que el servidor es iniciado con la opción –a. Usualmente esto significa que necesitará agregar –a (sin quota) al final de la líneas en el fichero */etc/inetd.conf* que inicia con el servidor.

Para sistemas operativos que no trabajan con el *inetd* deberá consultar la información agregada en los HOW-TO.

Por defecto el servidor actúa como un servidor *FTP* regular (sin extensiones).

Escribimos en el *shell*:

```
[root@lhost] /# kill –1 ‘ps t”?” | grep inetd’".
```

Para el soporte del servidor para la creación de archivos .tar.Z de directorios, necesitamos el *GNU tar* instalador (recuerde poner una copia en el *FTP* anónimo).

Si necesita una copia, se consigue desde el servidor *prep..ai.mit.edu* en el directorio */pub/gnu*.

Copiamos el programa compress en nuestro directorio *~ftp/bin/compress*, copiamos el programa *ls* en *~ftp/bin/*. Si nuestro sistema operativo usa librerías shared y estos programas no tienen un link estático, necesitaremos duplicar las librerías shared usadas al sitio correcto de uso. Donde *~ftp* es el directorio donde instalamos el *ftp*.

Usamos el programa *~ftp/bin/ckconfig* para dar el primer paso para poner varios archivos de configuración del *ftpd* como son: *ftpconversions*, *ftpusers* y *ftpgroups*.

Colocamos algunos ejecutables si queremos que los usuarios puedan ejecutarlos (con el comando *SITE EXEC*) en el directorio *_SITE_EXEC*. Debemos tener cuidado de colocar este comando activo ya que puede ocasionar devastaciones al nivel de seguridad.

Volvemos a ejecutar ckconfig para asegurarnos que todos los archivos soportados se encuentran correctamente instalados.

4.3 CONFIGURACIÓN PARA ENJAULAR EL SERVICIO FTP CON WU-FTP.

Generalmente la configuración de estos servicios se lleva a cabo desde el super-servidor inetd y archivos de configuración del sistema operativo, así pues la configuración de este servicio es conjunta con archivos del sistema y los de configuración del servicio en paralelo, esto quiere decir que un cambio en uno puede afectar a otros más aunque no lo parezca, por esta razón si usted no es un usuario avanzado debe delegar estos pasos de configuración al sistema ó programas que se pueden encargar de ello así corra riesgo la seguridad del servicio.

Para protoger el servicio de un atacante se debe separar el directorio de servicio del raíz y no asignarle un *shell* al *ftp* para que los usuarios ejecuten comandos; para esto editamos el archivo */etc/shell* y vemos si se encuentra la línea */dev/null*, que no le permitirá al atacante entrar por el *FTP* al sistema:

```
#/etc/shells
#
/bin/bash.132
/bin/sh
/bin/ash
```

```
/bin/bsh
/bin/tcsh
/bin/csh
/dev/null
```

Ahora editamos el *archivo /etc/passwd*, y añadimos un punto (.) en la línea tal como se observa:

De:

```
ftpadmin:x:502:502::/home/ftp/ftpadmin/:/dev/null
```

A:

```
ftpadmin:x:502:502::/home/ftp/./ftpadmin/:/dev/null
```

Esto hace que ese directorio sea un nuevo *filesystem*, dividiendo los directorios para evitar intrusiones. Convirtiendo ese directorio en raíz.

Lo que esencialmente estamos haciendo es crear el esqueleto de un directorio *root* con los componentes necesarios y archivos binarios, contraseñas, etc. Que

nuestro sistema operativo usará cuando los usuarios accedan al sistema por este servicio. Hay que notar que si usa la sentencia `--enable-ls option` durante la compilación del paquete los directorios */home/ftp/bin*, y */home/ftp/lib* no son requeridos desde que *WU-FTP* tiene su propia función *ls*. Nosotros accederemos al viejo método copiando el ls del sistema y las librerías relativas a este comando.

Los siguientes pasos son necesarios para enjaular el servicio FTP. Primero creamos los directorios que usaremos con los siguientes comandos desde el root:

```
[root@lhost ] /# mkdir /home/ftp/dev
[root@lhost ] /# mkdir /home/ftp/etc
[root@lhost ] /# mkdir /home/ftp/bin
[root@lhost ] /# mkdir /home/ftp/lib
```

Como dijimos los dos últimos comandos sólo son usados si no usamos la opción `--enable-ls` la compilación.

Cambiamos los permisos de los directorios a 0511 por razones de seguridad, el comando *chmod* debe asegurarse que los directorios contenidos puedan ser

leídos y ejecutados por el root y ejecutados para los grupos de usuarios y todos los usuarios:

```
[root@lhost ] /# chmod 0511 /home/ftp/dev/
[root@lhost ] /# chmod 0511 /home/ftp/etc/
[root@lhost ] /# chmod 0511 /home/ftp/bin
[root@lhost ] /# chmod 0511 /home/ftp/lib
```

Como dijimos los dos últimos comandos sólo son usados si no usamos la opción `--enable-ls` la compilación.

Copiamos el archivo /bin/ls al directorio */home/ftp/bin* y cambiamos los permisos a 0111. No queremos que los usuarios puedan modificar los binarios:

```
[root@lhost ] /# cp /bin/ls /home/ftp/bin
[root@lhost ] /# chmod 0111 /bin/ls /home/ftp/bin/ls
```

Este paso no es requerido si no usamos la opción `--enable-ls` la compilación. Buscamos las dependencias de librerías del programa de *Linux ls*:

```
[root@lhost ] /# ldd /bin/ls
```

Requerido solamente si no usamos `--enable-ls option`.

```
libc.so.6 => /lib/libc.so.6 (0x00125000)
/lib/ld-linux.so.2 =7gt; /lib/ld-linux.so.2 (0x00110000)
```

Copiamos las librerías identificadas en nuestro nuevo directorio *lib* bajo el directorio */home/ftp*:

```
[root@lhost ] /# cp /lib/libc.so.6 /home/ftp/lib/
[root@lhost ] /# cp /lib/ld-linux.so.2 /home/ftp/lib/
```

Solo es requerido si no usamos `--enable-ls option`.

Estas librerías son necesarias para que ls trabaje, estos pasos son solo importantes si no usamos `--enable-ls` ya que *WU-FTP*, implementa la capacidad de ejecutar este comando internamente (desde la versión 2.6.0).

Creamos nuestro archivo */home/ftp/dev/null*:

```
[root@lhost ] /# mknod /home/ftp/dev/null c 1 3
[root@lhost ] /# chmod 666 /home/ftp/dev/null
```

Copiamos los archivos de contraseñas de grupos en el directorio */home/ftp/etc*.

Esto podría no ser igual a los usuarios reales. Por esta razón debemos remover a todos aquellos que no son usuarios de ese servicio, excepto al super usuario *root*, en ambos archivos *passwd* y *group*:

```
[root@lhost ] /# cp /etc/passwd /home/ftp/etc/
[root@lhost ] /# cp /etc/group /home/ftp/etc/
```

Editamos el archivo */home/ftp/etc/passwd* y eliminamos todas las entradas excepto la del super usuario *root* y los usuarios *FTP*. Es importante que el archivo *passwd* contenga por lo menos entradas como estas:

```
root:x:0:0:root:/:/dev/null
ftpadmin:x:502:502::/ftpadmin/:/dev/null
```

El directorio home de los usuarios dentro de este archivo *passwd* debe ser cambiada al reflejo de el directorio raíz *FTP*, el nombre de los usuarios debe ser cambiada a */dev/null*.

A continuación editamos el archivo de los grupos */home/ftp/etc/group* y eliminamos todas las entradas excepto las del super-usuario root y todos los usuarios *FTP* admitidos. El archivo de grupo puede corresponder a un grupo normal, por lo menos debe tener las entradas siguientes:

```
root:x:0:root
ftpadmin:x:502:
```

Ahora colocaremos estos archivos inmutables para asegurarnos que no cambiaran de la noche a la mañana:

```
[root@lhost ] /# cd /home/ftp/etc/
[root@lhost ] /# chattr +i passw
[root@lhost ] /# cd /home/ftp/etc/
[root@lhost ] /# chattr +i group
```

4.4 CONFIGURACION.

Como la mayoría de programas en plataformas tipo Unix la configuración se basa en archivos tipo texto ubicados en el directorio */etc* que más adelante explicaremos.

Para ejecutar un servidor *FTP*, los siguientes archivos son requeridos y deben ser creados ó copiados a los directorios apropiados en el servidor.

Copiar los archivos:

- `ftpaccess` en el directorio */etc/*
- `ftpusers` en el directorio */etc/*
- `ftphosts` en el directorio */etc/*
- `ftpgroups` en el directorio */etc/*
- `ftpconversion` en el directorio */etc/*

- `ftp` en el directorio */etc/pam.d/*

- `ftpd` en el directorio */etc/logrotate.d/*

4.4.1 Configurando el archivo /etc/ftphost. El archivo */etc/ftphosts* es usado para definir a los usuarios cuando estos tienen acceso a un login ó por el contrario se les niega el acceso.

Creamos el archivo *ftphosts*, con el comando `touch /etc/ftphosts` y añadimos las siguientes líneas como ejemplo:

```
#
# archivo ftphosts
#.135
allow ftpadmin 208.164.186.1 208.164.186.2 208.164.186.4
#aceptamos al usuario ftpadmin desde los IP de los
#servidores.
deny ftpadmin 208.164.186.5
#denegamos si intenta ingresar desde la IP
```

```
#correspondiente.
#fin de archivo.
```

Ahora cambiamos los permisos por defecto a 600:

```
[root@lhost ] /# chmod 600 /etc/ftphosts
```

4.4.2 Configurando el archivo /etc/ftpusers. Este archivo especifica a los usuarios que no podrán conectarse a su servidor FTP.

Creamos el archivo *ftpusers*:

```
touch /etc/ftpusers
```

Añadimos en este archivo los siguientes usuarios por razones de seguridad:

```
root
bin
daemon
adm
```

```
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
```

Ahora, cambiamos estos permisos por defectos a 600:

```
[root@lhost ] /# chmod 600 /etc/ftpusers
```

4.4.3 Configurando el archivo /etc/ftpconversions. Este archivo contiene instrucciones que permite comprimir archivos en demanda antes de ser transferidos.

Editamos el archivo *ftpconversions* y añadimos en este archivo las siguientes líneas:

```
:.Z: : :/bin/compress -d -c
%s:T_REG|T_ASCII:O_UNCOMPRESS:UNCOMPRESS.136
: : :.Z:/bin/compress -c
%s:T_REG:O_COMPRESS:COMPRESS
:.gz: : :/bin/gzip -cd
%s:T_REG|T_ASCII:O_UNCOMPRESS:GUNZIP
: : :.gz:/bin/gzip -9 -c %s:T_REG:O_COMPRESS:GZIP
: : :.tar:/bin/tar -c -f - %s:T_REG|T_DIR:O_TAR:TAR
: : :.tar.Z:/bin/tar -c -Z -f -
%s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+COMPRESS
: : :.tar.gz:/bin/tar -c -z -f -
%s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+GZIP
: : :.crc:/bin/cksum %s:T_REG::CKSUM
: : :.md5:/bin/md5sum %s:T_REG::MD5SUM
```

Ahora cambiamos los permisos por defectos a 600:

```
[root@lhost ] /# chmod 600 etc/ftpconversions
```

4.4.4 Configurando el archivo /etc/pam.d/ftp. Configure este archivo para que use la autenticación pam creando el archivo */etc/pam.d/ftp* y añadiendo las siguientes líneas:

```
#%PAM-1.0
auth required /lib/security/pam_listfile.so item=user \
\sense=deny file=/etc/ftpusers onerr=succeed
auth required /lib/security/pam_pwdb.so shadow nullok
auth required /lib/security/pam_shells.so
account required /lib/security/pam_pwdb.so
session required /lib/security/pam_pwdb.so
```

4.4.5 Configurando el archivo /etc/logrotate.d/ftpd. Este archivo automáticamente rota los archivos de log cada semana, para que este archivo realice su función debe crearlo en la ubicación /etc/logrotate.d/ftpd y añade las siguientes líneas:

```
/var/log/xferlog {
# ftpd doesn't handle SIGHUP properly
nocompress
```

```
}
```

4.4.6 Configurando el servicio para el uso de Tcp-wrappers. Para configurar el servicio ftpd para el uso de *Tcp-wrappers* podría habilitar ó deshabilitar el servicio. Antes de la ejecución inetd lee esa configuración desde el fichero de configuración por defecto, que es */etc/inetd.conf*.

Editamos dicho fichero y añadimos ó verificamos la existencia de la siguiente línea:

```
ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -l -a
```

Actualizamos nuestro archivo inetd.conf para enviar una señal *SIGHUP*, *killall –HUP inetd*, después de haber añadido la línea en el archivo:

```
[root@lhost ] /# killall -HUP inetd
```

Editamos el archivo /etc/hosts.allow y añadimos por ejemplo la siguiente línea:

```
in.ftpd: 192.168.1.4 ftp.sistemas.cutb.edu.co
```

Donde reconocerá el IP del cliente 192.168.1.4 con el nombre cliente.sistemas.cutb.edu.co como servidor del cliente nuestro servidor *FTP*.

4.5 ASEGURANDO EL SERVIDOR FTP.

Es importante configurar el archivo *ftpusers*, ya que en este fichero aparecerán los usuarios que no deben tener acceso al servidor. En este archivo se deben incluir como mínimo las siguientes entradas:

```
root, bin, daemon, adm, lp, sync, shutdown, halt, mail, news,
uucp, operator, games, nobody
```

Además de todos aquellos usuarios por defecto que las distribuciones adicionan, estas cuentas se encuentran en el archivo */etc/passwd*.

Para deshabilitar *anonymous FTP*, debemos eliminar el usuario anonymous desde el archivo de contraseñas y verificar que el paquete *anonftp-version.i386.rpm* no este instalado en nuestro sistema.

Para remover el usuario *ftp* de nuestro archivo de contraseña, usamos el siguiente comando:

```
[root@lhost ] /# userdel ftp
```

Para verificar que el paquete rpm de *anonymous* no se encuentre instalado en nuestro sistema Linux, usamos el siguiente comando:

```
[root@lhost ] /# rpm -q anonftp
package anonftp is not installed
```

Por defecto el servidor *wu-ftpd* se asigna privilegios de montaje a todos los usuarios. Esto hace que cualquier usuario remoto pueda colocar archivos en el servidor. Para asegurar estabilidad en el servicio no podemos acceder que los usuarios ingresen archivos en los directorios *bin*, *etc*, *dev*, y *lib* del directorio */home/ftp*. En nuestro archivo */etc/ftpaccess* debemos tener los usuarios para el directorio */home/ftp* y ellos no pueden acceder a cualquier área fuera de esta estructura de directorio, pero en los casos donde los permisos deben ser negados para los directorios */home/ftp/, /home/ftp/bin*, */home/ftp/etc*, */home/ftp/dev*, y */home/ftp/lib* que deben ser especificados en el archivo */etc/ftpaccess*.

Editamos el archivo */etc/ftpaccess* y añadimos las siguientes líneas para negar los servicios de mntaje en zonas prohibidas:

```
# We don't want users being able to upload into these areas.
upload /home/ftp/* / no
upload /home/ftp/* /etc no
upload /home/ftp/* /dev no
upload /home/ftp/* /bin no
upload /home/ftp/* /lib no
```

Las dos últimas líneas son requeridas si no compilamos con la opción `–enabled-ls`.

Las líneas anteriores niegan el montaje en los directorios de nuestro directorio raíz para este servicio, */, /etc*, */dev*, */bin* y */lib* de nuestra estructura */home/ftp*.

4.6 EL ARCHIVO ESPECIAL .NOTAR.

Cuando habilitamos sentencias tar de directorios por este servicio, debemos asegurarnos que ocurra un final de aplicación donde no pueden usar el comando

tar en todas las areas donde este comando no sea permitido. Para hacer esto, creamos un archivo especial *.notar* en cada directorio y en el directorio *FTP*:

```
[root@lhost ] /# touch /home/ftp/.notar
[root@lhost ] /# touch /home/ftp/etc/.notar
[root@lhost ] /# touch /home/ftp/dev/.notar
[root@lhost ] /# touch /home/ftp/bin/.notar
[root@lhost ] /# touch /home/ftp/lib/.notar
[root@lhost ] /# chmod 0 /home/ftp/.notar
[root@lhost ] /# chmod 0 /home/ftp/etc/.notar
[root@lhost ] /# chmod 0 /home/ftp/dev/.notar
[root@lhost ] /# chmod 0 /home/ftp/bin/.notar
[root@lhost ] /# chmod 0 /home/ftp/lib/.notar
```

Los directorios */home/ftp/lib* y */home/ftp/lib* solo son requeridos sí no compilamos el fuente con la opción `--enable-ls`.

El tamaño de cero (0) del archivo puede confundir algunos clientes *web* y *proxies FTP*, pero para arreglar este problema añadimos líneas a nuestro archivo */etc/ftpaccess:*

```
noretrieve .notar
```

El parámetro `noretrieve` del servidor *wu-ftpd* accede ó niega una transferencia de los archivos ó directorios sentenciados. Esto podría ser una excelente idea para prevenir la descarga de archivos de los directorios como *bin*, *etc*, *dev* y *lib* en el directorio de nuestro *FTP /home/ftp* con el comando noretrieve en nuestro archivo */etc/ftpaccess*:

```
# Podemos prevenir descargas con noretrieve.
noretrieve /home/ftp/etc
noretrieve /home/ftp/dev
#las siguientes solo son requeridas si no se compilo
#con –enable-ls
noretrieve /home/ftp/bin
noretrieve /home/ftp/lib
```

4.7 ARCHIVOS INSTALADOS.

Los archivos que instala la aplicación FTP en nuestro sistema son:

```
/etc/pam.d/ftp
```

```
/etc/logrotate.d/ftpd
/etc/ftpaccess
/etc/ftpconversions
/etc/ftpgroups
/etc/ftphosts
/etc/ftpusers
/home/ftp/
/usr/bin/ftpcount
/usr/bin/ftpwho
/usr/man/man1/ftpcount.1
/usr/man/man1/ftpwho.1
/usr/man/man5/ftpaccess.5
/usr/man/man5/ftphosts.5
/usr/man/man5/ftpconversions.5
/usr/man/man5/xferlog.5
/usr/man/man8/ftpd.8
/usr/man/man8/ftpshut.8
/usr/man/man8/ftprestart.8
/usr/sbin/in.ftpd
/usr/sbin/ftpshut
/usr/sbin/ckconfig
```

```
/usr/sbin/ftprestart.140

/usr/sbin/xferstats

/usr/sbin/wu.ftpd

/usr/sbin/in.wuftpd

/var/log/xferlog.141
```

5. EL SERVICIO DE CORREO

Uno de los usos más comunes de las redes informáticas desde sus orígenes ha sido el correo electrónico. Empezó siendo un simple servicio que copiaba un fichero de una máquina a otra, y lo añadía al fichero mailbox (buzón de correo) del destinatario.

Básicamente, en esto sigue consistiendo el *mail* (correo electrónico), aunque una red en continuo crecimiento Con sus complejas necesidades de encaminado y su volumen de mensajes siempre en aumento ha hecho necesario un esquema más elaborado. Varios estándares de intercambio de correo han sido diseñados. Los nodos conectados a la Internet cumplen uno recogido en el *RFC* 822, complementado en algunos *RFCs* que describen un método independiente de la máquina para transferir caracteres especiales, y similares.

Mucho se ha discurrido recientemente sobre el “correo multi-media”, que tiene que ver con incluir imágenes y sonido en los mensajes de correo.

Otro estándar, *X.400*, ha sido definido por el *CCITT*. Una gran cantidad de programas para transporte de correo han sido implementados para sistemas. Uno

de los más conocidos es el sendmail, de la Universidad de Berkeley, que se usa en diversas plataformas. El autor original fue *Eric Allman*, que está trabajando activamente en el equipo sendmail de nuevo.

5.1 ¿QUÉ ES UN MENSAJE DE CORREO?

Un mensaje de correo consta de un contenido (*body*), que es el texto que ha escrito el remitente, y datos especiales que especifican el destinatario o destinatarios, el medio de transporte, etc., de manera similar a lo que aparece en el sobre de una carta ordinaria.

Estos datos administrativos se clasifican en dos categorías; en la primera categoría están los datos que son específicos del medio de transporte, como son las direcciones del remitente y del destinatario. A esto se le llama el sobre (*envelope*). Puede ser modificado por el software de transporte a medida que el mensaje es transmitido.

La segunda variedad es cualquier dato necesario para la manipulación del mensaje, que no es propio de ningún mecanismo de transporte, como es la línea del encabezado en la que indicamos el tema del mensaje (*Subject*), la lista de

todos los destinatarios, y la fecha en la que se envió el mensaje. En muchas redes, se ha convertido en un estándar incluir estos datos al comienzo del mensaje, formando lo que se denomina encabezado del mensaje (*mail header*). Se separa del contenido del mensaje (*mail body*) por una línea en blanco.

La mayoría del software para transporte de correo que se usa en el mundo usa un formato de encabezado definido en el *RFC* 822.

Su propósito original era especificar un estándar para usar en la *ARPANET*, pero dado que fue diseñado para ser independiente del entorno de uso, ha sido fácilmente adaptado a otras redes, incluyendo muchas basadas en UUCP.

Pero *RFC* 822 es sólo el máximo denominador común. Otros estándares más recientes han sido concebidos para dar respuesta a las crecientes necesidades como pueden ser, por ejemplo, encriptación de datos, soporte de conjuntos de caracteres internacionales, y extensiones de correo multimedia (*multi-media mail extension*, *MIME*).

En todos esos estándares, el encabezado consiste en varias líneas, separadas por caracteres de retorno de carro. Cada línea consiste en un nombre de campo, que

comienza en la columna uno, y el campo en sí, separados por dos puntos (:) o un espacio. El formato y la semántica de cada campo varía dependiendo del nombre del mismo. Un campo del encabezado se puede continuar más allá de una línea, si la línea siguiente comienza con un TAB. Los campos pueden aparecer en cualquier orden. Un encabezado de correo típico puede ser algo así:

```
From yahoo.com!ora.com!andy Wed Jan 18 13:21:21 2000
Return-Path: <yahoo.com!ora.com!andy>
Received: from yahoo.com by operamail.com with uucp
          (Smail3.1.28.1 #6) id m0pptLt-000023aB; Wed, 18 Jan
     00 13:21 MET DST
Received: from ora.com (ruby.ora.com) by yahoo.com with smtp
     (Smail3.1.28.1 #28.6) id <m0pqrr-0008qhC>; Tue 17 Jan 00
01:23 MEST
Received: by ruby.ora.com (8.6.8/8.6.4) id RAA26438; Tue 17
Jan 00 04:15 –0400
Date: Tue, 17 Jan 2000 04:15:49 –0400
Message-Id: <200001171956.PAA07787@ruby>
From: andy@ora.com (Andy Omar)
To: okir@operamail.com
Subject: Re: Su mensaje acerca de los RPC
```

Usualmente, todos los campos del encabezado necesarios son generados por la interfaz con el servidor de correo que usted use como *elm*, *pine*, *mush*, *outlook*, *netscape* o *mailx*.

Algunos, sin embargo, son opcionales, y pueden ser añadidos por el usuario. elm por ejemplo, permite editar parte del encabezado del mensaje. Otros campos son añadidos por el software de transporte de correo. Una lista de campos de encabezado comunes y su significado se da a continuación:

- From: Contiene la dirección de correo electrónico del remitente, y posiblemente el “nombre real”. Un verdadero zoológico de formatos distintos se usa aquí.
- To: Ésta es la dirección de email del destinatario.

- Subject: Describe el contenido del mensaje en pocas palabras.

- Date: La fecha en la que el mensaje fue enviado.

- Reply-To: Especifica la dirección a la que el remitente desea que el destinatario le conteste. Esto puede ser útil sí se tienen varias direcciones, pero si desea

recibir la mayor parte del correo sólo en aquélla que se usa más a menudo. Este campo es opcional.

- Organization: La organización que posee la máquina desde la que se ha enviado el mensaje. Si la máquina usada es la suya propia no incluya este campo, o bien indique “privado” o cualquier trivialidad sin sentido. Este campo es opcional.

- Message-ID: Una cadena generada por el transporte de correo en el sistema remitente. Es única para cada mensaje.

- Received: Cada nodo que procesa su correo (incluyendo las máquinas del remitente y el destinatario) insertan este campo en el encabezado, dando el nombre del nodo, una identificación de mensaje, hora y fecha a la que lo recibieron, de qué nodo procede, y qué software de transporte ha sido usado. Esto se hace así para que usted pueda conocer la ruta que su mensaje ha seguido, y pueda protestar a la persona responsable si algo ha ido mal.

- X-cualquier-cosa: Ningún programa relacionado con el correo debe protestar sobre cualquier encabezado que comience con *X*-. Esto se usa para implementar características adicionales que aún no han sido incluidas en un *RFC*, o que no lo

serán nunca. Esto se usa, por ejemplo, en la lista de correo de los Activistas de Linux, donde el canal a usar se selecciona con el campo de encabezado *X-Mn-Key*.

La única excepción a esta estructura es la primera línea. Comienza con la palabra clave *From* seguida de un espacio en blanco, en vez de dos puntos. Para distinguirlo del campo ordinario *From*: se suele denotar como *From_*. Contiene la ruta que ha seguido el mensaje, escrita al estilo ruta bang de *UUCP* (explicado más adelante), la hora y la fecha en que fue recibido por la última máquina que lo ha procesado, y una parte opcional que especifica desde que máquina ha sido recibida. Como este campo es regenerado por cada sistema que procesa el mensaje, algunas veces queda incluido en los datos del sobre.

El campo *From_* continúa existiendo por compatibilidad con procesadores de correo antiguos, pero no se usa demasiado en la actualidad excepto por algunas interfaces de usuario de correo que se basan en él para marcar el comienzo de un mensaje en el buzón del usuario. Para evitar problemas potenciales con líneas del contenido del mensaje que comiencen también con "*From*", se ha convertido en práctica común distinguir este último caso precediéndolo de un "".

5.2 ¿CÓMO SE REPARTE EL CORREO?

Generalmente, usted escribirá su correo usando una interfaz de correo como *mail* o *mailx*; u otros más sofisticados como *elm*, *mush*, o *pine*. Estos programas se denominan agentes de usuario de correo (*mail user agents*), o *Musa* para abreviar.

Sí usted envía un mensaje de correo, el programa interface en la mayoría de los casos se lo pasará a otro programa para que lo transmita. Este programa se denomina el agente de transporte de correo *(mail transport agent*), o *MTA*.

En algunos sistemas hay agentes de transporte de correo distintos para envíos locales o lejanos; en otros hay sólo un *MTA*. El comando para envíos lejanos se denomina usualmente rmail, el otro se denomina lmail (si existe). Un envío local de correo es, por supuesto, algo más que añadir el mensaje al buzón del destinatario.

Usualmente el *MTA* local entenderá como usar alias (definir direcciones locales de destinatarios que dirigen a otras direcciones), y como usar redirecciones en inglés forwarding, dirigir el correo de un usuario a otra dirección). También, los mensajes que no pudieron ser enviados deben ser usualmente devueltos (*bounced*), al remitente junto con algún mensaje de error.

Para envíos lejanos, el software de transporte usado depende del tipo de enlace. Si el correo debe enviarse a través de una red que usa *TCP/IP*, se usará normalmente *SMTP*. *SMTP* son las siglas de Simple Mail Transfer Protocol, o Protocolo Simple de Trasferencia de Correo que se define en el *RFC* 788 y *RFC* 821. *SMTP* usualmente conecta con la máquina del destinatario directamente, negociando la transferencia del mensaje con el demonio *SMTP* del otro lado.

En redes tipo *UUCP*, el correo usualmente no es enviado directamente, sino que es redirigido hasta su destino a través de un conjunto de máquinas intermedias. Para enviar un mensaje a través de un enlace *UUCP*, el *MTA* remitente ejecutará usualmente *rmail* en la máquina intermedia usando *uux*, y suministrándole el mensaje en la entrada estándar. Dado que esto se hace para cada mensaje por separado, puede producir una carga considerable de trabajo en un nodo procesador de correo grande, además de inundar las colas *UUCP* con cientos de pequeños mensajes que ocupan una cantidad de disco desproporcionada.

Por esto algunos *MTAs* permiten recopilar varios mensajes de un sistema remoto en un solo fichero de lotes. El fichero de lotes contiene los comandos *SMTP* que el nodo local ejecutaría normalmente si usara una conexión *SMTP* directa. A esto se

le llama *BSMTP*, o batched *SMTP* (*SMTP* por lotes). El lote es suministrado al programa *rsmtp* o *bsmtp* en el sistema remoto, que procesará la entrada como si una conexión *SMTP* normal hubiera ocurrido.

5.3 DIRECCIONES DE CORREO

Para el correo electrónico, una dirección consiste en, al menos, el nombre de la máquina que maneja el correo del destinatario, y una identificación de usuario reconocida por ese sistema. Puede ser el nombre de acceso del destinatario, pero puede ser también cualquier otra cosa.

Otros esquemas de direcciones, como el *X.400*, usan un conjunto más general de "atributos" que se utilizan para buscar la máquina del destinatario en un servidor de directorio *X.500*.

La forma en que se interpreta un nombre de máquina, es decir, a qué nodo va a llegar finalmente nuestro mensaje, y cómo combinar este nombre con el nombre de usuario del destinatario depende enormemente de la red en la que nos encontremos. Los nodos en la Internet siguen el estándar *RFC* 822, que requiere una notación `usuario@máquina.dominio`, donde `máquina.dominio` es el

nombre de *dominio* totalmente cualificado (Fully Qualified Domain Name, o *FQDN*) de la máquina.

El objeto que aparece entre medias se denomina signo "*at*". (N. del T. de la preposición inglesa "at", que significa "en"). Dado que esta notación no indica la ruta hasta la máquina de destino, sino que da el nombre (único) de dicha máquina, a esto se le suele llamar una dirección absoluta.

En el entorno UUCP original, la forma predominante era `ruta!máquina!usuario`, donde ruta describía una secuencia de máquinas a través de las cuales debía viajar el mensaje para llegar la máquina, su destino final. Esta construcción se llama la notación ruta *bang*, porque un signo de exclamación se denomina coloquialmente "*bang*".

Hoy en día muchas redes basadas en *UUCP* han adoptado el *RFC* 822, y entenderán ese tipo de dirección. Estos dos tipos de direcciones no se mezclan muy bien. Supongamos una dirección `máquinaA!usuario@máquinaB`. No queda claro sí el signo '@' tiene precedencia sobre la ruta o viceversa:

Hemos de enviar el mensaje a la *máquinaB*, que lo enviará a *máquinaA!usuario*, o debe ser enviado *máquinaA*, que lo redirigirá a *usuario@máquinaB*?.

Las direcciones que mezclan diferentes tipos de operadores de dirección se denominan direcciones *híbridas*. El más notorio es el ejemplo anterior. Se resuelve usualmente dándole precedencia al signo '@' sobre la ruta.

En el ejemplo anterior, esto significa enviar el mensaje a la *máquinaB* primero. De todos modos, hay una forma de especificar rutas acordes con *RFC* 822:

```
$<$@máquinaA,@máquinaB:usuario@máquinaC$>$
```

Denota la dirección de usuario en *máquinaC*, indicando que se debe llegar a la *máquinaC* a través de *máquinaA* y *máquinaB* (en ese orden). Este tipo de dirección se suele llamar una dirección *route-addr* (de *route*, *ruta* y *address*, *dirección*). Y también existe el operador de dirección '*%*':

```
usuario%máquinaB@máquinaA
```

Será enviado primero a *máquinaA*, que sustituirá el signo de tanto por ciento que se encuentre más a la derecha en la expresión (en este caso el único) por un

signo ‘@’. La dirección quedará ahora *usuario@máquinaB*, y el gestor de correo redirigirá alegremente el mensaje a la máquinaB que lo entregará a usuario.

Este tipo de dirección se suele denominar a veces como “Ye Olde ARPANET Kludge”, (“La Vieja Chapuza de ARPANET”) y su uso está desaconsejado. Aun así muchos agentes de transporte de correo generan este tipo de direcciones. Otras redes tienen más formas distintas de expresar direcciones. Las redes basadas en el protocolo *DECnet*, por ejemplo, usan dos signos dos puntos como separador, dando lugar a direcciones como *máquina::usuario.13.3* Finalmente, el estándar *X.400* usa un esquema totalmente distinto, describiendo a un destinatario por un conjunto de pares *atributo-valor*, como país u organización.

En *FidoNet*, cada usuario se identifica por un código como *2:320/204.9*, que consiste en cuatro números que denotan la *zona* (2 es Europa), *red* (320 es París y Banlieve), *nodo* (el repetidor/BBS local), y *punto* (el PC del usuario). Las direcciones Fidonet se pueden traducir a *RFC* 822: la anterior se escribiría *Thomas.Quinot@p9.f204.n320.z2.fidonet.org*.

¿No se ha dicho antes que los nombres de dominio son fáciles de recordar?. Hay algunas implicaciones al usar esos tipos diferentes de direcciones que serán

descritas a lo largo de las próximas secciones. De todos modos, en un entorno *RFC* 822 raramente se usará otra cosa que direcciones absolutas como *usuario@máquina.dominio*.

5.4 EL ENCAMINADO EN INTERNET DEL CORREO.

En la Internet, depende enteramente del nodo de destino que se realice algún encaminado específico de correo. El comportamiento por defecto consiste en enviar el mensaje al nodo de destino buscando su dirección IP, y dejando el encaminado en sí de los datos a la capa IP de transporte. La mayoría de los nodos usualmente querrán que todo el correo entrante se dirija a un servidor de correo fácilmente accesible que sea capaz de procesar todo ese tráfico, y que distribuirá ese correo localmente. Para anunciar ese servicio, el nodo publica el llamado campo *MX* para su dominio local en la base de datos *DNS*.

MX significa *Mail Exchanger* (Intercambiador de correo) y básicamente quiere decir que el servidor va a actuar como un redistribuidor de correo para todas las máquinas de este dominio. Los campos *MX* también pueden usarse para manipular el trafico dirigido a máquinas que no están ellas mismas conectadas a la Internet, como redes *UUCP* o redes corporativas que contienen información

confidencial. Los campos MX también tienen una preferencia asociada. Es un entero positivo.

Si existen varios intercambiadores de correo para una máquina, el agente de transporte de correo intentará enviar el mensaje al intercambiador con menor valor de preferencia, y sólo si este falla probará uno con mayor valor. Si el nodo local es él mismo un intercambiador de correo para la dirección de destino, debe no redirigir los mensajes a cualquier máquina *MX* que tenga un valor de preferencia mayor que el suyo propio: ésta es una forma segura de evitar bucles de correo. Supongamos que una organización, digamos Foobar Inc., quiere que todo su correo sea manipulado por su máquina llamada *mailhub*. Entonces tendrán un campo *MX* como el siguiente en su base de datos *DNS*:

```
foobar.com          IN   MX   5    mailhub.foobar.com
```

Esto anuncia que `mailhub.foobar.com` es un intercambiador de correo para foobar.com con un valor de preferencia de 5.

Una máquina que desee enviar un mensaje a *joe@greenhouse.foobar.com* buscará el registro *DNS* de foobar.com, y encontrará el campo *MX* apuntando

hacia mailhub. Si no hay ningún *MX* con un valor de preferencia menor que 5, el mensaje será enviado a *mailhub*, que lo entregará a greenhouse.

Lo anterior es sólo un esbozo de cómo funcionan los campos *MX*. Para más información sobre encaminado de correo en la Internet, por favor consulte el *RFC* 974.

5.5 DISEÑO DEL SERVIDOR DEL CORREO

Sin importar el *MTA* a usar como administrador de la red ó de este servicio, debe de colocar políticas y prevendas para este servicio, estas normas deben ir paralelas a las políticas tomadas de otros servicios, así como las políticas que manejen la empresa. Los datos a continuación pueden dar una imagen de un diseño de servicio de correo:

- Número de usuarios.

- Razón social del servicio (correo gratuito, empresarial, comunicación interna, educativo, etc.).

- Zona de cobertura del servicio.

- Conexiones con el servicio.

- Capacidad del hardware.

- Privilegios de cada usuario.

Con estos datos podemos especificar la quota para cada usuario, restricciones y configuración de todo este mecanismo.

Debemos saber que este es el servicio más usado y promovido por el protocolo TCP/IP, y hoy por hoy se ha convertido en un medio de comunicación viable para el ahorro de papeleria en comunicaciones dentro de una red.

Este servicio actualmente puede prestarse de distintas maneras y puede acarrear servicios que antes ni soñabamos tener con una cuenta de mail. Por todas estas razones usted como administrador de la red debe tener en cuenta que es el medio de comunicación que usted tiene con sus usuarios ¡cuidelo!.

5.6 EL MTA QMAIL.

Qmail es un Agente de Transporte de Correo (MTA, *Mail Transport Agent* en inglés) para sistemas operativos tipoUNIX. Se trata de un sustituto completo para el sistema *sendmail* que se suministra con los sistemas operativos UNIX. *qmail* utiliza el *Simple Mail Transfer Protocol* (*SMTP*, Protocolo Simple de Transferencia de Correo) para intercambiar mensajes con los *MTA* (Agentes de Transporte de Correo) de otros sistemas.

Atención: Su nombre es qmail, no Qmail. qmail está formado por conjunto de programas que se integran en un paquete seguro. Se ofreció un premio de $1,000.00 a de todo aquel capaz de demostrar lo contrario, que quedó desierto.

Puede bajar *qmail 1.03*, *RPMs* disponibles (qmail.org), (Bruce Guenter) y redistribuirlo libremente. Puede tener una visión de conjunto del funcionamiento de qmail, de cómo está organizado. qmail está a prueba de los efectos de año 2000.

Una muestra de quiénes están usando qmail:

- Hotmail (correo saliente).

- USA.net (correo saliente).

- Yahoo!.

- Network Solutions.

- listserv.acsu.buffalo.edu (un concentrador listserv enorme, utiliza qmail desde 1996).

- XOOM.com.

- onelist.com (que actualmente está negociando la compra de egroups, otro servidor gratuito de listas de correo).

- USWest.net (proveedor del oeste de los EE UU).

- RIPE.
- Matchlogic.

- Algonet.se (proveedor sueco).

- gmx.de (proveedor alemán).

- Teleport (el mayor proveedor de Oregón).

- NetZero (proveedor gratuito).

- Critical Path («outsourcing» de correo con 7 Millones de buzones).

- PayPal/Confinity.

- Red Hat (listas de correo).

- Hypermart.net, casema, Rediffmail, Topica, vuurwerk.nl...

Su sistema operativo probablemente incluya *Sendmail* como *MTA*, las siguientes son razones para desplazarlo. Algunas de las ventajas de *qmail* sobre los *MTA* suministrados con el sistema son:

- Seguridad *qmail* se diseño pensando en una seguridad alta. *Sendmail* arrastra una larga historia plagada de serios problemas de seguridad. Cuando se escribió

*Sendma*il, la Red era un lugar mucho más amigable. Todo el mundo conocía a todo el mundo, y apenas había necesidad de diseñar y programar pensando en alta seguridad.

Hoy en día Internet es un entorno mucho más hostil para los servidores de red. El autor de *Sendma*il, Eric Allman, ha hecho un gran trabajo al ensamblar el programa, pero nada que se aleje de una redefinición del diseño podrá conseguir seguridad *re*al.

- Rendimiento *qmail* paraleliza el envío de correo, llevando a cabo de forma predeterminada hasta 20 entregas simultáneas de correo.

- Fiabilidad una vez que *qmail* ha aceptado un mensaje, garantiza que no se perderá. *qmail* soporta también un nuevo formato de bandeja de correo que funciona con seguridad *incluso en NFS* sin recurrir al bloqueo de ficheros.

- Simplicidad *qmail* es más compacto y pequeño que cualquier otro MTA de características equivalentes.

5.7 INSTALACION DE QMAIL

La instalación de este programa puede ser llevada a cabo de dos maneras por paquetes ó compilando los fuentes, como siempre explicaremos la segunda opción, por lo que dice el dicho no hay nada como lo hecho en casa. Antes que nada debemos observar los requerimientos para observar que funcionará correctamente, los requerimientos principales son:

- Aproximadamente 10 Mb de espacio libre en el área de compilación, y durante el proceso de compilación. Después de la compilación, puede liberar todo el espacio excepto 4 Mb, si elimina los ficheros objeto.

- Un sistema de desarrollo en C completo y en funcionamiento, con un compilador, los ficheros de cabecera del sistema, y las bibliotecas. Las instrucciones de compilación le mostrarán cómo saber si dispone de los componentes requeridos.

- Varios megabytes para los binarios, la documentación y los ficheros de configuración.

- Espacio en disco suficiente para la cola de correo. Los sistemas pequeños para un único usuario precisan solamente un par de megabytes libres. Los servidores grandes pueden necesitar un par de gigas.

- Un sistema operativo compatible. La mayor parte de las variantes de UNIX son válidas. Véase README en el árbol de código fuente para una lista de versiones compatibles conocidas.

- Se recomienda encarecidamente un servidor de nombres de dominio (DNS). Sin él, *qmail* sólo puede enviar a sistemas remotos que estén en su fichero de configuración smtproutes.

- Conexiones a red adecuadas. *qmail* se diseñó para sistemas con buena conexión, así que es probable que no quiera usarlo para un servidor de listas de correo en una conexión telefónica de 28.8k. El paquete serialmail se diseñó para hacer que *qmail* fuese más compatible con sistemas con conexiones más lentas.

Luego de ver las especificaciones seguimos con la descarga de los ficheros de instalación, la versión más reciente puede ser descargada de los siguientes tres sitios:

- ftp://koobera.math.uic.edu/www/software/qmail-1.03.tar.gz.

- ftp://koobera.math.uic.edu/www/software/ucspi-tcp-0.84.tar.gz.

- ftp://koobera.math.uic.edu/www/daemontools/daemontools-0.61.tar.gz

Debemos descomprimir los ficheros descargados mueva los archivos al directorio en el que quiera trabajar. Una buena elección es */usr/local/src/* y en este caso puede usar */usr/local/src/qmail* para los tres paquetes:

```
mkdir -p /usr/local/src/qmail
mv *.tar.gz /usr/local/src/qmail<newline>
```

Ya tiene los tres paquetes en */usr/local/src/qmail*, de manera que ya puede descomprimirlos.

```
cd /usr/local/src/qmail
gunzip qmail-1.03.tar.gz
tar xvf qmail-1.03.tar
gunzip ucspi-tcp-0.84.tar.gz
tar xvf ucspi-tcp-0.84.tar.150
```

```
gunzip daemontools-0.61.tar.gz
tar xvf daemontools-0.61.tar
rm *.tar
```

Llegados aquí, tendría que haber tres subdirectorios llamados *qmail-1.03*, *ucspitcp- 0.84* y *daemontools-0.61*. Cámbiese al directorio *qmail-1.03* y comencemos.

5.7.1 Creación de los directorios. Puesto que el programa de instalación de *qmail* crea los subdirectorios según se necesitan, sólo es preciso crear el directorio principal de *qma*il:

```
mkdir /var/qmail
```

Y pasemos a la sección siguiente:

Nota: Si desea que algunos de los ficheros de *qmail* esté en en un directorio distinto a */var*, puede conseguirlo creando vínculos simbólicos bajo el directorio */var/qmail* que apunten a la localización elegida. Por ejemplo, puede conseguirse una distribución más homogenea haciendo:

```
mkdir /var/qmail
ln -s /usr/man /var/qmail/man
mkdir /etc/qmail
ln -s /etc/qmail /var/qmail/control
ln -s /usr/sbin /var/qmail/bin
```

5.7.2 Creacion de usuarios y grupos. Debe dar de alta el grupo qmail y los usuarios qmail antes de compilar nada:

```
[root@lhost] /# groupadd nofiles
[root@lhost] /# useradd -g nofiles -d /var/qmail/alias alias
[root@lhost] /# useradd -g nofiles -d /var/qmail qmaild
[root@lhost] /# useradd -g nofiles -d /var/qmail qmaill
[root@lhost] /# useradd -g nofiles -d /var/qmail qmailp
[root@lhost] /# groupadd qmail
[root@lhost] /# useradd -g qmail -d /var/qmail qmailq
[root@lhost] /# useradd -g qmail -d /var/qmail qmailr
[root@lhost] /# useradd -g qmail -d /var/qmail qmails
```

5.7.3 Compilando el programa. Después de haber añadido los grupos necesarios puede compilar el programa de la siguiente manera:

```
[root@lhost] / # make setup check
```

5.7.4 Post instalación. Como ha podido ver, qmail no requiere esencialmente de configuración alguna previa a su compilación. No tendrá que recompilarlo nunca, a menos que quiera modificar el directorio propio de qmail, los nombres de usuarios, o sus uids. qmail permite muchísimas adiciones o retoques posteriores a su instalación de forma muy sencilla y fácil. Si le importa cómo salude su máquina a otra vía *SMTP*, por ejemplo, puede escribir la línea apropiada en */var/qmail/control/smtpgreeting*.

Pero todo esto es opcional; si control/smtpgreeting no existe, qmail establecerá algo razonable en su defecto. No debería preocuparse demasiado por la configuración en estos momentos, siempre podrá regresar y afinar las cosas posteriormente.

Hay una gran excepción. DEBE decirle a qmail el nombre de su sistema. Bastará con que invoque el script *config-fast*:

```
[root@lhost] /# ./config-fast su.nombre.demaquina.completo
config-fast
```

Pondrá su.nombre.demaquina.completo en *control/me*. También lo establecerá en *control/locals* y *control/rcpthosts*, para que qmail acepte correo destinado a *su.nombre.demaquina.completo*.

Puede en su lugar utilizar el programa config, que averiguará su nombre de sistema vía DNS:

```
[root@lhost] /# ./config
```

`config` también consultará sus direcciones IP locales vía DNS a fin de decidir para qué sistemas aceptar correo.

¿Por qué no permite qmail realizar dichas consultas al vuelo? Esto se debe a una decisión deliberada en el diseño.qmail realiza todas sus funciones locales (reescritura de cabeceras, comprobar si un destinatario es local, etc) sin tener que utilizar la red. La idea es que qmail siga aceptando y entregando correo incluso si su conexión a la red cae.

5.8 INSTALACION DE USCI-TCP

Pero, ¿porque usar *usci-tcp*? *tcpserver* y *tcpclient* son herramientas de fácil uso, para la línea de comandos, para elaborar aplicaciones cliente-servidor *TCP*. tcpserver espera conexiones entrantes, y ejecuta un programa de su elección por cada conexión. Su programa recibe variables de entorno mostrando el nombre del sistema local y remoto, las direcciones IP y los puertos.

tcpserver Ofrece un límite de concurrencia, para protegerle de agotar el máximo número de procesos o memoria disponible. Cuando se gestionan 40 conexiones simultaneas (por defecto), tcpserver va posponiendo fluidamente la aceptación de nuevas conexiones.

tcpserver proporciona asimismo controles de acceso *TCP*, similares a *host.allow* del paquete tcp-wrappers/tcpd pero mucho más rápidos. Las reglas de control de acceso son compiladas en un formato precalculado con cdb de modo que puedan gestionar miles de máquinas distintas.

Este paquete incluye una herramienta, recordio que monitoriza todas las entradas/salidas de un servidor.

tcpclient establece una conexión *TCP* y ejecuta un programa de su elección. Establece las mismas variables de entorno que tcpserver.

Este paquete incluye varios ejemplos de clientes elaborados sobre tcpclient who@, date@, finger@, http@, tcpcat, y mconnect. tcpserver y tcpclient se ciñen al interfaz de programación *UCSPI*, *UNIX* Client-Server Program Interface, empleando el protocolo *ucspi-tcp*.

Existen herramientas *UCSPI* para distintos tipos de redes.
Otras herramientas *TCP* de línea de comandos.

La interfaz *ucspi-tcp* es producto del refinamiento de la de tcpserver/tcpclient de mi paquete clientserver de 1991, que sustituyó a la interfaz attachport/authtcp de mi paquete auth de 1989.

ucspi-tcp compite ahora con varios programas más:

- *inetd*, un servidor *TCP* utilizable sólo por *root*, distribuido por todos los vendedores de UNIX.

- *xinetd*, un sustituto de inetd;

- El cliente *mconnect* que se incluye como parte de SunOS;

- El programa *socket*;

- *faucet* y *hose*, parte del paquete netpipes;

- El programa *netcat* que también soporta UDP.

Muchos servidores están sustituyendo inetd por tcpserver, por varias razones:

- inetd es poco fiable bajo grandes cargas. Suspende el servicio durante 10 minutos si recibe «demasiadas» conexiones en un minuto.

- inetd no proporciona un mecanismo de control de consumo de recursos efectivo. Agotará toda su memoria si ofrece algún tipo de servicio que sea popular.

- inetd tiene problemas para los aumentos súbitos de actividad. Su «lista de tareas pendientes» para *listen()* abarca 5 o 10 como máximo, y no puede incrementarse.

Anteriormente ya había desempaquetado los .tar.gz de *qmai*l, *ucpsi-tcp* y *daemontool*s. En nuestro ejemplo los habíamos desempaquetado en el directorio */usr/local/src/qmail*. Ahora cambiemos al directorio de *ucpsi-tcp*:

```
cd /usr/local/src/qmail/ucspi-tcp-0.84
```

Sí modificó `conf-cc` y `conf-ld` necesitará hacer los mismos cambios en este directorio. Luego ejecute:

```
make
make setup check
```

Eso es todo. *ucpsi-tcp* estará instalado.

5.9 INSTALACION DE DAEMONTOOLS

Cambiémonos al directorio de compilación de *daemontools*:

```
cd /usr/local/src/qmail/daemontools-0.61
```

Una vez más, si modificó `conf-cc` y `conf-ld` durante las compilaciones de *qmail* y *ucspi-tcp*, necesitará hacer los mismos cambios en este directorio. Luego ejecute:

```
make
make setup check
```

Compruebe la compilación siguiendo las instrucciones que aparecen en http://pobox.com/~djb/daemontools/install. (Disponibles en castellano en http://www.es.qmail.org/software/autor/daemontools/)

5.10 INICIO DEL SERVICIO

El directorio */var/qmail/boot* contiene ejemplos de guiones de arranque de *qmail* para diferentes configuraciones:

/var/spool/mail frente a *$HOME/Mailbox*, uso de *procmail* o de *dot-forward*, y varias combinaciones de ambos.

Tómese la libertad de examinarlos, pero en este caso y para nuestra instalación, usaremos los siguientes:

```
#!/bin/sh
# Usamos la salida estándar para llevar un registro
# Usamos el control/defaultdelivery de qmail-local como forma
predeterminada
# para entregar mensajes
exec env - PATH="/var/qmail/bin:$PATH" \
qmail-start "‘cat /var/qmail/control/defaultdelivery‘"
```

Utilice su editor para crear el archivo anterior */var/qmail/rc*, y luego ejecute las siguientes órdenes:

```
chmod 755 /var/qmail/rc
mkdir /var/log/qmail
```

Llegados a este punto, tendrá que decidir el modo predeterminado de entrega para los mensajes que no se entregan según las instrucciones de un fichero qmail.

La siguiente tabla apunta algunas de las opciones más comunes:

Formato Buzón	**Nombre**	**Localización**	**Entrega Por Defecto**	**Comentarios**
mbox	Mailbox	$HOME	./Mailbox	lo más usual, lo soportan más clientes
maildir	Maildir	$HOME	./Maildir/	Más fiable, lo soportan menos clientes
Mbox	nombreusuario	/var/spool/mail	V*erINSTALL.vs*	Buzón

			m	tradicional UNÍS

Tabla 7. Opciones para entrega de mensajes del correo

Para seleccionar su tipo de buzón por defecto, introduzca el valor *EntregaPorDefecto* de la tabla en */var/qmail/control/defaultdelivery*. Por ejemplo para seleccionar la entrega estándar de *qmail* en *Mailbox*, introduzca:

```
echo ./Mailbox >/var/qmail/control/defaultdelivery.
```

Nota: *defaultdelivery* no es un fichero de control estándar de *qma*il. Es una característica del fichero */var/qmail/rc* visto más arriba.

5.10.1 Ficheros de inicio del sistema. Si tuviera que ejecutar manualmente el guión */var/qmail/rc*, *qmail* sólo se iniciaría en parte. Pero queremos que *qmail* se inicie automáticamente cada vez que el sistema arranque, y que *qmail* se pare cada vez que el sistema se detenga.

Esto se consigue creando un guión de inicio/parada como el siguiente:

```
#!/bin/sh
PATH=/var/qmail/bin:/usr/local/bin:/usr/bin:/bin
export PATH
case "$1" in
start)
echo -n "Iniciando qmail: svscan"
cd /var/qmail/supervise
env - PATH="$PATH" svscan &
echo $! > /var/run/svscan.pid
echo "."
;;
stop)
echo -n "Deteniendo qmail: svscan"
kill ‘cat /var/run/svscan.pid‘
echo -n " qmail"
svc -dx /var/qmail/supervise/*
echo -n " logging"
svc -dx /var/qmail/supervise/*/log
echo ".".155
;;
stat)
```

```
cd /var/qmail/supervise
svstat * */log
;;
doqueue|alrm)
echo "Enviando una señal ALRM a qmail-send."
svc -a /var/qmail/supervise/qmail-send
;;
queue)
qmail-qstat
qmail-qread
;;
reload|hup)
echo "Enviando una señal HUP a qmail-send."
svc -h /var/qmail/supervise/qmail-send
;;
pause)
echo "Congelando qmail-send"
svc -p /var/qmail/supervise/qmail-send
echo "Congelando qmail-smtpd"
svc -p /var/qmail/supervise/qmail-smtpd
;;
```

```
cont)
echo "Reanudando qmail-send"
svc -c /var/qmail/supervise/qmail-send
echo "Reanudando qmail-smtpd"
svc -c /var/qmail/supervise/qmail-smtpd
;;
restart)
echo "Reiniciando qmail:"
echo "* Deteniendo qmail-smtpd."
svc -d /var/qmail/supervise/qmail-smtpd
echo"* Enviando a qmail-send la señal SIGTERM y reiniciando."
svc -t /var/qmail/supervise/qmail-send
echo "* Reiniciando qmail-smtpd."
svc -u /var/qmail/supervise/qmail-smtpd
;;
cdb)
tcprules /etc/tcp.smtp.cdb /etc/tcp.smtp.tmp < /etc/tcp.smtp
chmod 644 /etc/tcp.smtp*
echo "Releído /etc/tcp.smtp."
;;
help)
```

```
cat << HELP
stop -- detiene el servicio de correo (conexiones smtp
rehusadas, nada sale afuera)
start -- inicia el servicio de correo (conexiones smtp
aceptadas, el correo puede salir)
pause -- congela temporalmente el servicio de correo (se
aceptan conexiones, pero no sale nada afuera)
cont -- continúa con el servicio de correo antes congelado
stat -- muestra el estado del servicio de correo
cdb -- reconstruye el fichero cdb de tcpserver para smtp
restart -- detiene y reinicia smtp, envía qmail-send una
señal TERM y lo reinicia
doqueue -- envía a qmail-send una señal de ALRM,
reprogramando los mensajes salientes para su entrega
reload -- envía a qmail-send una señal de HUP, leyendo de
nuevo locals y virtualdomains
queue -- muestra el estado de la cola de correo
alrm -- lo mismo que doqueue
hup -- lo mismo que reload
HELP
;;
```

```
*)
echo "Uso: $0
{start|stop|restart|doqueue|reload|stat|pause|cont|cdb|queue|
help}"
exit 1
;;
esac
exit 0
```

Este guión también puede conseguirse por web en http://Web.InfoAve.net/~dsill/qmail-script-dt61.txt.

Nota: Si encuentra que *qmail* se detiene poco después de reiniciar el sistema, puede anteponer la orden `supervise` en la sección de `start` del guión con `nohup`. Por ejemplo:

```
nohup env - PATH="$PATH" svscan &
```

Cree el guión usando su editor de texto u obténgalo de Internet con su navegador y luego instálelo en el directorio *init.d* de su sistema, que debería estar en una de las localizaciones siguientes:

- /etc/init.d

- /sbin/init.d

- /etc/rc.d/init.d

Llame al guión *qmai*l. También tendrá que hacer un vínculo simbólico al guión en algunos de los directorios *rc*. Estos directorios se nombran *rcN.d*, donde *N* es el nivel de ejecución *(runleve*l) al que se aplican. Las interioridades del árbol de directorios del inicio quedan más allá de la finalidad de este documento.

Sí no le bastan estas instrucciones simplificadas, consulte la documentación de sus sistema. Su directorios rc estarán probablemente en uno de estos sitios:

- /etc

- /sbin

- /etc/rc.d

Para crear los vínculos simbólicos, ejecute las siguientes ordenes, cambiando *RCDIR* por la localización de los directorios *rc* de su sistema:

```
ln -s ../init.d/qmail RCDIR/rc0.d/K30qmail
ln -s ../init.d/qmail RCDIR/rc1.d/K30qmail
ln -s ../init.d/qmail RCDIR/rc2.d/S80qmail
ln -s ../init.d/qmail RCDIR/rc4.d/S80qmail
ln -s ../init.d/qmail RCDIR/rc5.d/S80qmail
ln -s ../init.d/qmail RCDIR/rc6.d/K80qmail
```

Nota: los números del paso anterior son en gran medida dependientes del sistema, pero en cierto modo flexibles. Si Sendmail está instalado, la ejecución de la orden `find RCDIR -name "*sendmail" -print` le dará los números que deben funcionar para su sistema. Ahora cree los directorios de supervise para los servicios *qma*il:

```
mkdir -p /var/qmail/supervise/qmail-send/log
mkdir -p /var/qmail/supervise/qmail-smtpd/log
chmod +t /var/qmail/supervise/qmail-send
chmod +t /var/qmail/supervise/qmail-smtpd
```

Cree el fichero */var/qmail/supervise/qmail-send/run*:

```
#!/bin/sh
exec /var/qmail/rc
```

Cree el fichero */var/qmail/supervise/qmail-send/log/run*:

```
#!/bin/sh
exec /usr/local/bin/setuidgid qmaill /usr/local/bin/multilog
t /var/log/qmail
```

Cree el fichero */var/qmail/supervise/qmail-smtpd/run*:

```
#!/bin/sh
QMAILDUID=’id -u qmaild’
NOFILESGID=’id -g qmaild’
exec /usr/local/bin/softlimit -m 2000000 \
/usr/local/bin/tcpserver -v -p -x /etc/tcp.smtp.cdb \
-u $QMAILDUID -g $NOFILESGID 0 smtp /var/qmail/bin/qmail-
smtpd 2>&1
```

Cree el fichero */var/qmail/supervise/qmail-smtpd/log/run*:

```
#!/bin/sh
exec /usr/local/bin/setuidgid qmaill /usr/local/bin/multilog
t /var/log/qmail/smtpd
```

Haga ejecutables los ficheros que correspondientes:

```
chmod 755 /var/qmail/supervise/qmail-send/run
chmod 755 /var/qmail/supervise/qmail-send/log/run.158
chmod 755 /var/qmail/supervise/qmail-smtpd/run
chmod 755 /var/qmail/supervise/qmail-smtpd/log/run
```

Luego configure los directorios para los archivos de registro:

```
mkdir -p /var/log/qmail/smtpd
chown qmaill /var/log/qmail /var/log/qmail/smtpd
```

Haga ejecutable el guión de inicio y hágale un enlace simbólico en un directorio de su ruta (sustituya la correcta localización de su directorio *rc* en las dos siguientes líneas):

```
chmod 755 /etc/init.d/qmail
ln -s /etc/init.d/qmail /usr/local/sbin
```

Permita que los usuarios locales inyecten correo por medio de SMTP:

```
echo ’127.:allow,RELAYCLIENT=""’ >> /etc/tcp.smtp
/usr/local/sbin/qmail cdb
```

5.10.2 Detenga y desactive el MTA (agente de transporte de correo) instalado. Si bien es posible ejecutar simultáneamente *qmail* y el *MTA* existente, que probablemente sea Sendmail, no lo recomiendo a menos que sepa lo que esta haciendo. Y francamente, si está leyendo estas líneas, entonces es que no sabría lo que estaba haciendo).

Si su *MTA* existente es Sendmail podrá detenerlo ejecutando el guión de inicio con el argumento stop. Por ejemplo, alguno de los siguientes debe funcionar:

```
/etc/init.d/sendmail stop
/sbin/init.d/sendmail stop
/etc/rc.d/init.d/sendmail stop
```

Sí no encuentra un guión de inicio de *init.d/sendmail*, puede localizar el *PID* de sendmail's PID usando `ps -ef j grep sendmail` o bien `ps waux j grep sendmail` y detenerlo utilizando:

```
kill PID-de-sendmail
```

Sí su *MTA* no es Sendmail compruebe la documentación para el correcto procedimiento de detención.

También debería pensar en eliminar completamente de su sistema el *MTA* anterior. Al menos, desactive el guión init.d de manera que no intente arrancar de nuevo cuando el sistema reinicie.

Para Red Hat Linux, la desinstalación de Sendmail se consigue así:

```
rpm -e --nodeps sendmail
```

Por fin, sustituya cualquier */usr/lib/sendmail* existente con la versión de *qma*il:

```
mv /usr/lib/sendmail /usr/lib/sendmail.old # ignore los errores
mv /usr/sbin/sendmail /usr/sbin/sendmail.old # ignore los errores
chmod 0 /usr/lib/sendmail.old /usr/sbin/sendmail.old # ignore los errores.159
ln -s /var/qmail/bin/sendmail /usr/lib
ln -s /var/qmail/bin/sendmail /usr/sbin
```

Ya estamos muy cerca de poder iniciar *qma*il. El último paso es crear algunos alias del sistema.

5.10.3 Inicie *qmail.* Por fin, ya puede arrancar *qma*il:

```
/usr/local/sbin/qmail start
```

5.11 CONFIGURACION

Ya ha instalado *qma*il, bien mediante el método del .tar.gz del código fuente, o mediante uno de los paquetes que se autocompilan. Esta sección contiene

información que le será necesaria al administrador del correo o del sistema para configurar *qmail* de manera que funcione de la manera que desean.

5.11.1 Ficheros de configuración. Todos los ficheros de configuración de sistema de *qma*il, con la excepción de los ficheros *.qmail* en *~alias*, están en */var/qmail/control*. La página man de *qmail-control* contiene una tabla como la que aparece con el nombre de los Ficheros de control.

5.11.2 Nombres de servidor múltiples. Si su sistema tiene más de un nombre, es decir, direcciones del formato *usuario@servidor1.proveedor.com* se pueden escribir también como *usuario@proveedor.com* o bien *usuario@correo.proveedor.com*, entonces tiene que indicárselo a *qmail* para que sepa qué direcciones debe entregar localmente, y qué mensajes debe aceptar para sistemas remotos.

Para hacer esto, añada todos los nombres a dos ficheros de control:

- rcpthosts, que indica a qmail-smtpd que acepte correo dirigido a estos servidores, y también.

- locals, que indica a qmail-send que las direcciones sobre estas máquinas han de entregarse localmente.

5.11.3 qmail-users. qmail-users es un sistema de asignación de direcciones a usuarios. Bajo */var/qmail/users* reside una serie de ficheros de configuración. El fichero assign es una tabla de asignaciones. Hay dos tipos de asignaciones: sencilla y comodín.

Nota: *assign* contiene una serie de asignaciones, una por línea, seguida por una línea que contiene un punto (.). Si modifica *assign* manualmente, no olvide la línea final con el punto.

Asignación sencilla, Una asignación sencilla tiene este aspecto:

```
=dirección:usuario:uid:gid:directorio:guión :extensión:
```

Esto significa que los mensajes recibidos para dirección se ejecutarán como el usuario usuario con los uid y gid (identificadores de usuario y grupo) especificados, y que el fichero *directorio/.qmail* guión extensión especificará cómo se han de entregar los mensajes.

Asignación por comodines, Una asignación por comodines presenta este aspecto:

```
+prefijo:usuario:uid:gid:directorio:guión:prefijo:
```

Lo que quiere decir que los mensajes recibidos para direcciones en la forma *prefijoresto* se ejecutarán como el usuario usuario, con el *uid* y el *gid* especificados, y el fichero *directorio/.qmailguiónprefijoresto* especificará la forma en que se entregarán los mensajes.

Programas de qmail-user, mail-user tiene dos programas de ayuda: *qmail-newu* y *qmail-pw2u*. *qmail-newu* procesa el fichero *assign* y genera un fichero de base de datos constante (*CDB*) llamado cdb en */var/qmail/users*. *CDB* está en un formato binario al qmail-spawn puede acceder rápidamente, incluso cuando hay cientos de asignaciones. *qmail-pw2u* convierte la base de datos de usuarios del sistema, */etc/passwd*, en una serie de asignaciones válidas para assign. *qmail-pw2u* utiliza una serie de ficheros que modifican las reglas de transformación de usuarios:

- include: usuarios por incluir.

- exclude: usuarios por excluir.

- mailnames: nombres de correo alternativos para los usuarios.

- subusers: usuarios adicionales manejados por un usuario, con una extensión .qmail opcional.

- append: asignaciones misceláneas.

Nota: si utiliza *qmail-pw2u* no olvide volver a ejecutar qmail-pw2u y qmail-newu cada vez que añada usuarios, los elimine o les cambie su UID o GID.

5.12 SERVIDORES POP E IMAP

qmail incluye un servidor POP, qmail-pop3d, pero no se configura ni instala como parte del proceso de instalación de *qma*il. También puede usar uno de los restantes servidores POP e IMAP disponibles, aunque la mayor parte de ellos se escribieron para sendmail y pueden necesitar de algún trabajo extra para que funcionen con *qma*il.

5.12.1 qmail-pop3d. *qmail-pop3d* es el servidor *POP* que se incluye con *qmail*. Es un servidor *POP* excelente y muchos sitios con *qmail* lo utilizan. Es modular y soporta múltiples esquemas de autentificación a través de módulos de autentificación alternativos.

Nota: *qmail-pop3d* soporta únicamente buzones de correo de formato maildir, de manera que si tienen usuarios que se autentifiquen en el servidor *POP* y ejecuten Agentes de Usuario de Correo localmente, todos estos Agentes tienen que soportar *maildir*. Si todos sus usuarios leen su correo vía *POP*, entonces el formato *mailbox* en el server no es una limitación.

Arquitectura de *qmail-pop3d*, un servidor *qmail-pop3d* está compuesto por tres módulos:

- qmail-popup: obtiene el nombre de usuario y la contraseña

- checkpassword: autentifica el nombre de usuario y la contraseña

- qmail-pop3d: el demonio *POP*

Generalmente *qmail-popup* se ejecuta a través de *inetd* o *tcpserver*, a la escucha en el puerto 110, el puerto de *POP3*. Cuando se efectúa una conexión, pregunta por el nombre de usuario y la clave. Invoca entonces checkpassword que verifica el nombre de usuario y la contraseña y llama a su vez a *qmail-pop3d* en caso de que coincidan.

Instalación de *qmail-pop3d*:

- Instale y verifique completamente *qma*il. Si desea que todos los usuarios tengan buzones de correo que se puedan recoger mediante *POP*, asegúrese de que defaultdelivery tiene como valor *./Maildir/.* Si instaló el guión *qmail* de la sección de instalación, se configurará en control/defaultdelivery. En caso contrario, probablemente esté en */var/qmail/rc* en la línea de órdenes de *qmail-start*.

- Obtenga una copia del programa *checkpassword* de http://www.qmail.org/top.html#checkpassword. El *checkpassword* estándar, http://pobox.com/~djb/checkpwd.html, es una buena elección si no precisa de algo especialmente lujoso.

- Compile e instale checkpassword siguiendo las indicaciones. Asegúrese de instalarlo como */bin/checkpassword*.

- Para un servidor *POP* que tenga poco uso, añada una entrada en /etc/inetd.conf como la siguiente:

```
pop3 stream tcp nowait root /var/qmail/bin/qmail-popup qmail-
popup
hostname.domain /bin/checkpassword /var/qmail/bin/qmail-pop3d
Maildir
```

Nota: Algunos sistemas, en especial Red Hat Linux, no llaman al puerto de *POP3* por el nombre *pop3* Compruebe en el fichero /etc/services el nombre del servicio en el puerto 110. Verifique igualmente la página man de su inetd para asegurarse de que la entrada tiene el formato adecuado. La parte complicada es que algunos inetds precisan que el primer argumento del programa (en este caso *qmail-popup* sea el nombre del programa). Otros *inetds* precisan sólo los argumentos " reales ":

- *kill -HUP PID* de inetd para decirle a *inetd* que vuelva a leer */etc/inetd.conf*.

- Para un servicio de mayor acceso, utilice en su lugar *tcpserver*.

Para usar *tcpserver*, añada la siguiente línea al guión de inicio de *qmail (no inetd.conf*):

```
tcpserver -v -R 0 pop3 /var/qmail/bin/qmail-popup FQDN \
/bin/checkpassword /var/qmail/bin/qmail-pop3d Maildir 2>&1 |
\
/var/qmail/bin/splogger pop3d &
```

en donde *pop3* es el nombre del servicio *POP3* listado en */etc/services* y *FQDN* es el nombre de dominio completo calificado del servidor *POP* que está configurando, por ejemplo: *pop.ejemplo.net*.

5.12.2 qpopper. Si precisa de un demonio *POP* que funcione con buzones de correo de formato *mbox*, puede utilizar *qpopper* de Qualcomm. Vince Vielhaber tiene un parche, disponible en http://www.qmail.org/qpopper2.53.patch.tar.gz que le permite funcionar con buzones de correo en los directorios de cada usuario. qpopper está disponible en http://www.eudora.com/freeware/qpop.html.

5.13 MIGRACION DESDE EL SENDMAIL

Deberá eliminar sendmail antes de instalar qmail definitivamente. Podrá seguir los pasos:

- Localice a sendmail en sus scripts de inicio. Suele estar bien en */etc/rc* o */etc/init.d/sendmail*. Se invoca con algo como `sendmail -bd -q15m -q15m` significa que debe procesar la cola cada 15 minutos. Esta cifra puede ser diferente.

- Mate al demonio de sendmail. Deberá primero hacer un `kill -STOP` al demonio. Si hay procesos hijos en ejecución, deberá hacerles `kill -CONT`, esperar, `kill -STOP` otra vez, y repetir `ad nauseam`. Si no hay procesos hijos, `kill –TERM` y entonces `kill -CONT`.

- Compruebe si tiene mensajes esperando en la cola de sendmail, */var/spool/mqueue*. Si los tiene, tendrá que intentar procesarlos con `sendmail.bak -q` otra vez. Repítalo hasta que la cola esté vacía. Esto puede durar varios días.

➢ Elimine el bit *setuid* del binario de sendmail, para impedir que los usuarios puedan ganar privilegios extra a partir de los agujeros de seguridad de sendmail. El ejecutable puede estar en varias localizaciones:

```
# chmod 0 /usr/lib/sendmail
# chmod 0 /usr/sbin/sendmail
# chmod 0 /usr/lib/sendmail.mx
```

➢ Quite el binario de sendmail de enmedio:

```
# mv /usr/lib/sendmail /usr/lib/sendmail.bak
# mv /usr/sbin/sendmail /usr/sbin/sendmail.bak
```

He aquí como eliminar el sistema de entrega local de correo *binmail* de su sistema. No haga esto si ha configurado *qmail* para utilizar *binmail* para las entregas locales:

➢ Localice el binario de binmail en su sistema: suele estar en */usr/libexec/mail.local* si existe, en otro caso /bin/mail.

➢ Elimine los permisos del binario de *binmail*:

```
# chmod 0 /usr/libexec/mail.local
```

- Si el binario de binmail era */bin/mail* asegúrese de que mail invoca todavía a un gestor de correo utilizable. Bajo *SVR4*, puede que desee vincular mail a *mailx*.

- Comente la línea comsat en */etc/inetd.conf* y envíe un `kill -HUP` al proceso de su demonio *inetd*. `csh -cf '/var/qmail/rc &'` A sus scripts de inicio, de tal modo que los demonios qmail sean rearrancados siempre que el sistema sea reiniciado. Asegúrese de incluir el `&`.

Ponga el sucedáneo de sendmail que proporciona qmail disposición de los Agentes de Usuario de Correo:

```
# ln -s /var/qmail/bin/sendmail /usr/lib/sendmail
# ln -s /var/qmail/bin/sendmail /usr/sbin/sendmail
```

/usr/sbin puede no existir en su sistema.

Introduzca una entrada para *qmail-smtpd* en */etc/inetd.conf* (todo en una sola línea):

```
smtp stream tcp nowait qmaild /var/qmail/bin/tcp-env tcp-env
/var/qmail/bin/qmail-smtpd
```

Reinicie. (O haga un `kill -HUP` al *uid* de su proceso *inetd* y asegúrese de que sus demonios qmail están ejecutándose).

```
% telnet 127.0.0.1 25
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
220 domain ESMTP
helo dude
250 domain.164
mail 250 ok
rcpt 250 ok
data
354 go ahead
Subject: testing
This is a test.
.
250 ok 812345679 qp 12345
```

```
quit
221 domain
Connection closed by foreign host.
%
```

Busque el mensaje en su buzón. (Nota para programadores: la mayoría de los servidores *SMTP* necesitan más texto entre *MAIL* y *RCPT*. Ver RFC 821):

Prueba de remoto a local: envíese a sí mismo mensajes desde otra máquina. Busque el mensaje en su buzón.

Prueba de remoto a error: envíe algún mensaje desde otra máquina a *inexistente@sumaquina*. Busque el mensaje de vuelta en el buzón remoto.

Prueba de Agente de Usuario de Correo: intente enviar correo, primero a una cuenta local, luego a una cuenta remota, con su Agente de Usuario de Correo habitual.

Prueba de remoto a postmaster: envíe un mensaje desde otra máquina a *PoStMaStEr@domain*. Busque el mensaje en el buzón de alias, generalmente *~alias/Mailbox*..

¡Eso es todo! para comunicar instalaciones con éxito:

```
%( echo 'Nombre Apellido'; cat `cat SYSDEPS` ) | mail djb-qst@ cr.yp.to
```

Sustituya `Nombre Apellido` con sus datos.

5.14 GENERALIDADES.

Una instalación de *qmail* completa, en estado correcto de ejecución correcto, aunque mínima, debe seguir siempre los cuatro procesos siguientes:

- qmail-send ejecutándose como el usuario *qmails.*
- qmail-clean ejecutándose como el usuario *qmailq.*
- qmail-rspawn ejecutándose como el usuario *qmailr.*
- qmail-lspawn ejecutándose como el usuario *root.*

Dependiendo de su variante de UNIX, uno de los dos siguientes comandos listará estos procesos, y tal vez algunos más:

```
ps -ef | grep qmail
ps waux | grep qmail
```

Por ejemplo:

```
[pepin@lhost pepin]$ ps waux | grep qmail
pepin 2222 0.0 0.8 836 348 p4 S 10:25 0:00 grep qmail
qmaild 351 0.0 1.0 840 400 ? S N 12:43 0:00
/usr/local/bin/tcpserver -v
-x /etc/tcp.smtp.cdb -u 49491 -g 31314 0 smtp
/var/qmail/bin/qmail-smtpd
qmaild 2220 0.0 1.0 844 420 ? S N 10:25 0:00
/usr/local/bin/tcpserver -v
-x /etc/tcp.smtp.cdb -u 49491 -g 31314 0 smtp
/var/qmail/bin/qmail-smtpd
qmaill 365 0.0 0.8 748 344 ? S N 12:43 0:00 splogger qmail
qmailq 368 0.0 0.7 736 292 ? S N 12:43 0:00 qmail-clean
qmailr 367 0.0 0.6 732 272 ? S N 12:43 0:00 qmail-rspawn
```

```
qmails 350 0.0 0.8 776 336 ? S N 12:43 0:00 qmail-send
root 340 0.0 0.6 724 252 ? S N 12:43 0:00
/usr/local/sbin/supervise /var/supervise/
/var/qmail/rc
root 341 0.0 0.6 724 252 ? S N 12:43 0:00
/usr/local/sbin/supervise /var/supervise/
/usr/local/bin/tcpserver -v -x /etc/tcp.smtp
root 366 0.0 0.7 736 276 ? S N 12:43 0:00 qmail-lspawn
./Mailbox
[pepin@lhost pepin]$
```

Si ejecuta *qmail* o *qmail-smtpd* bajo supervise, como en el ejemplo anterior, debe ver igualmente estos procesos. Y si ejecuta *qmail-smtpd* bajo tcpserver, deberá ver un proceso padre *tcpserver* además de un proceso tcpserver adicional para cada conexión activa de *SMTP* de entrada. Si utiliza splogger (o *cyclog*) para manejar el registro de operaciones, tendrá un proceso splogger (o *cyclog*) ejecutándose como usuario *qmail*. Asimismo, si *qmail* está ocupado entregando mensajes localmente o remotamente, verá como máximo el número *concurrencylocal* de procesos de *qmail-local* y como máximo el número *concurrencyremote* de procesos de *qmail-remote*.

6. EL SERVICIO DE PROXY

Los servidores Proxy, contiene la capacidad de reguardar ancho de banda, proveer seguridad e incrementar la velocidad de navegación por la red.

Actualmente pocos programas para servicio de proxy están disponibles. Estos proxy tienen dos desventajas:

- Son comerciales.

- Ellos no soportan *ICP*, *ICP* es usado para intercambiar hints acerca de la existencia de *URLs* en caches cercanos.

Squid es la mejor opción para un servidor proxy-cache, ya que es robusto, gratuito y puede usar *ICP*.

Derivado del software de cacheo de la ARPA-funded Harvest research project, desarrollado por la National Laboratory for Applied Network Research y fundado por la National Science Foundation, *Squid* ofrece alto desempeño en cacheo de clientes web, y soporta objetos de datos de *FTP*, *Gopher* y http. El almacena estos

objetos en *RAM* manteniendo una robusta base de datos de objetos en el disco, tiene un acceso complejo de control de mecanismo y soporta el protocolo *SSL* para proveer conexiones seguras. En adición, puede heredar vínculos de otro *proxy Squid* para distribuir el cacheo de páginas.

SQUID es un potente y rápido servidor de caché de objetos. Hace proxy de sesiones *FTP* y *WWW*, lo que básicamente le da las propiedades de un servidor *FTP* y *WWW*, pero sólo lee y escribe ficheros dentro del directorio de su caché (o al menos eso esperamos), lo cual le hace relativamente seguro. Sería muy difícil utilizar *Squid* para comprometer el sistema, se ejecuta como usuario no root (generalmente '*nobody*'), de modo que no hay mucho de lo que preocuparse. La principal preocupación con *Squid* debería ser la configuración incorrecta. Por ejemplo, si se engancha *Squid* a la red interna (como suele ser el caso), y a Internet (de nuevo, bastante habitual), se podría utilizar para alcanzar hosts internos (incluso aunque estén utilizando direcciones IP no rutables). De aquí que sea muy importante la correcta configuración de *Squid*.

La forma más sencilla de asegurarse de que esto no ocurre es utilizar la configuración interna de *Squid* y enlazarlo sólo a los interfaces internos, no dejando que el mundo exterior lo intente utilizar como un proxy para llegar a tu

LAN interna. Además de esto, es una buena idea filtrarlo con el cortafuegos. Squid se puede utilizar como acelerador de *HTTP* (también conocido como proxy inverso), quizás se tenga un servidor NT en la red interna que se quiera compartir con el resto del mundo, en este caso las cosas se complican un poco a la hora de configurarlo, pero es posible hacerlo de una forma relativamente segura.

Afortunadamente *Squid* tiene buenas *ACL's* (listas de control de acceso) dentro del fichero *squid.conf*, lo cual te permite bloquear el acceso por nombres, *IP's*, redes, hora del día, día real (quizás se permita acceso ilimitado los fines de semana a gente que va a la oficina). Sin embargo, recuerda que cuanto más complicada sea la *ACL*, más lento será el *Squid* para responder a las peticiones.

6.1 INSTALACION.

En nuestra compilación y configuración noseotros debemos configurar *Squid* para que ejecute como un acelerador *httpd* para tener más eficiencia fuera de nuestro servidor web. En el modo acelerador, el servidor *Squid* actúa como un proxy caché de reversa, esto quiere decir que acepta las peticiones del cliente, sirviéndolo fuera del caché sí es posible, ó sirviendo las peticiones desde el servidor original por eso se le denomina proxy de reversa.

En los siguientes apartados mostraremos como configurar *Squid* como un servidor proxy-caché disponible para todos los usuarios de la red haciendo que *Squid* acceda a Internet.

Esta instalación asume:

- Los comandos son compatibles con UNIX.

- El directorio fuente es /var/tmp, otros son posibles.

- Todos los pasos en la instalación deberán ser con la cuenta del super usuario root.

- La versión Squid a usar es 2.3

- Los paquetes están disponibles en la página http://www.squid-cache.org/ squid-2.3.STABLE2-src.tar.gz.

Antes de descomprimir los archivos, es una buena idea hacer una lista de los archivos en el sistema antes de instalar el *Squid* y luego comparamos usando *dic* para detectar donde fueron ubicados. Ejecutamos del *shell*:

```
run find /* > Squid1 before and find /* > Squid2
```

Después de haber instalado el software y usamos:

```
diff Squid1 Squid2 > Squid-Installed
```

Para adquirir una lista de lo que cambio.

Para compilar, necesitamos descomprimir el *archivo tar.gz* que descargamos:

```
[root@lhost] /# cp squid-version.STABLEz-src.tar.gz /var/tmp
[root@lhost] /# cd /var/tmp
[root@lhost ]/tmp# tar xzpf squid-version.STABLEz-src.tar.gz
```

6.2 CONFIGURACION Y OPTIMIZACION ANTES DE LA INSTALACIÓN.

El servidor proxy *Squid* no puede ejecutarse como super usuario *root*, y por esta razón debemos crear una cuenta de usuario especial con no *shell* para ejecutar el servidor proxy *Squid*. Para ello ejecutamos:

```
[root@lhost] /# useradd -d /cache/ -r -s /dev/null squid
>/dev/null 2>&1
[root@lhost] /# mkdir /cache/
[root@lhost] /# chown -R squid.squid /cache/
```

Primero que todo, nosotros debemos añadir el usuario *squid* al archivo */etc/passwd*. Luego creamos el directorio */cache* si este directorio no existe.

Finalmente, cambiamos al propietario del directorio */cache* adjudicándoselo al usuario *squid*.

Usualmente nosotros no necesitamos comandos crear el directorio, porque si habíamos particionado el disco correctamente debemos tener una partición denominada */cache/* en nuestro disco.

A continuación modificaremos el archivo de configuración para la instalación para especificarle rutas al programa que usará al instalarse.

Nos desplazamos al nuevo directorio *Squid* y editamos el archivo *Makefile.in* y cambiamos la línea:

```
DEFAULT_ICON_DIR = $(sysconfdir)/icons
```

Por:

```
DEFAULT_ICON_DIR = $(libexecdir)/icons
```

Nosotros cambiamos la variable, *sysconfdir* por *libexecdir*. Con esta modificación, el directorio de iconos de Squid podrá ser localizado bajo el directorio the */usr/lib/squid*. Cambiamos las líneas:

```
DEFAULT_CACHE_LOG = $(localstatedir)/logs/cache.log
```

Por:

```
DEFAULT_CACHE_LOG = (localstatedir)/log/squid/cache.log
```

La línea:

```
DEFAULT_ACCESS_LOG = $(localstatedir)/logs/access.log
```

Por:

```
DEFAULT_ACCESS_LOG = $(localstatedir)/log/squid/access.log
```

La línea:

```
DEFAULT_STORE_LOG = $(localstatedir)/logs/store.log
```

Por:

```
DEFAULT_STORE_LOG = $(localstatedir)/log/squid/store.log
```

La línea:

```
DEFAULT_PID_FILE = $(localstatedir)/logs/squid.pid
```

Por:

```
DEFAULT_PID_FILE = $(localstatedir)/run/squid.pid.169
```

La línea:

```
DEFAULT_SWAP_DIR = $(localstatedir)/cache
```

Por:

```
DEFAULT_SWAP_DIR = /cache
```

La línea:

```
DEFAULT_ICON_DIR = $(sysconfdir)/icons
```

Por:

```
DEFAULT_ICON_DIR = $(libexecdir)/icons
```

Nosotros cambiamos las locaiones por defecto de los archivos *cache.log*, *access.log*, y store.log para ser localizados en el directorio */var/log/squid* directory.

Luego insertamos el archivo pid bajo el directorio */var/run* directory, y finalmente localizamos el directorio de iconos de *Squid* bajo */usr/lib/squid/icons* con la variable *libexecdir*.

Si sufrimos por limitaciones de memoria en el sistema, el desempeño de cache de Squid podría verse afectado. Para reducir este problema, podemos vincular *Squid* con una librería externa de alocación de memoria como *GNU malloc*. Para hacer esto seguimos los siguientes pasos:

Descargamos ó conseguimos el fichero a usar en este caso *malloc.tar.gz*, la página de descarga es http://www.gnu.org/order/ftp.html, ejecutamos desde el *shell* los siguientes comandos para descomprimir el fichero:

```
[root@lhost] /# cp malloc.tar.gz /var/tmp
[root@lhost] /# cd /var/tmp
[root@lhost ]/tmp# tar xzpf malloc.tar.gz
```

Compilamos e instalamos *GNU malloc* en nuestro sistema para ejecutar los siguientes comandos:

```
[root@lhost ]/tmp# cd malloc
```

```
[root@lhost ]/malloc# export CC=egcs
[root@lhost ]/malloc# make
```

Copiamos el archivo *libmalloc.a* a nuestro directorio de librerías del sistema y nos aseguramos de nombrarlo *libgnumalloc.a*:

```
[root@lhost ]/malloc# cp libmalloc.a /usr/lib/libgnumalloc.a
```

Copiamos el archivo *malloc.h* a el directorio include de nuestro sistema y nos aseguramos llamarlo *gnumalloc.h*:

```
[root@lhost ]/malloc# cp malloc.h /usr/include/gnumalloc.h
```

Con los archivos *libgnumalloc.a* y *gnumalloc.h* instalados en nuestro sistema, *Squid* podrá detectar automáticamente durante el tiempo de compilación y lo podemos usar para optimizar el desempeño del caché.

6.3 COMPILANDO Y OPTIMIZANDO EL SQUID.

Regresamos al nuevo directorio Squid y ejecutamos desde la terminal:

```
CC="egcs" \
CFLAGS="-O9 -funroll-loops -ffast-math -malign-double -mcpu=
pentiumpro   –march=pentiumpro   -fomit-frame-pointer   -fno-
exceptions"\
./configure \
--prefix=/usr \
--exec-prefix=/usr \
--bindir=/usr/sbin \
--libexecdir=/usr/lib/squid \
--localstatedir=/var \
--sysconfdir=/etc/squid \
--enable-delay-pools \
--enable-cache-digests \
--enable-poll \
--disable-ident-lookups \
--enable-truncate \
--enable-heap-replacement
```

Esto especifica la instalación del *Squid* con las siguientes características:

- Use un espacio de repetición del Squid para delimitar el ancho de banda usada por los usuarios.

- Use caché Digest para acelerar el tiempo de respuesta y utilización de la red.

- Activo poll() instancia de select() para preferir lo seleccionado.

- Deshabiltado ident-lookups para remover el código que desempeña Ident, RFC 931, lookups y reduce la posibilidad de un DoS (negación de servicio).

- Activo truncate para mantener el desempeño cuando removemos los archivos de caché.

- Usa la opción heap-replacement de *Squid* para tener la opción de escoger de varios algoritmos de caché, instancia de el algoritmo estándar *LRU* para mayor desempeño.

Ahora compilaremos e instalaremos *Squid* en el servidor:

```
[root@lhost ]/squid-2.3.STABLE2# make -f makefile
[root@lhost ]/squid-2.3.STABLE2# make install
```

```
[root@lhost ]/squid-2.3.STABLE2# mkdir -p /var/log/squid
[root@lhost ]/squid-2.3.STABLE2# rm -rf /var/logs/
[root@lhost ]/squid-2.3.STABLE2# chown squid.squid
/var/log/squid/
[root@lhost ]/squid-2.3.STABLE2# chmod 750 /var/log/squid/
[root@lhost ]/squid-2.3.STABLE2# chmod 750 /cache/
[root@lhost ]/squid-2.3.STABLE2# rm -f /usr/sbin/RunCache
[root@lhost ]/squid-2.3.STABLE2# rm -f /usr/sbin/RunAccel.171
[root@lhost ]/squid-2.3.STABLE2# strip /usr/sbin/squid
[root@lhost ]/squid-2.3.STABLE2# strip /usr/sbin/client
[root@lhost ]/squid-2.3.STABLE2# strip
usr/lib/squid/dnsserver
[root@lhost ]/squid-2.3.STABLE2# strip /usr/lib/squid/unlinkd
[root@lhost ]/squid-2.3.STABLE2# strip
/usr/lib/squid/cachemgr.cgi
```

El comando `make -f` compilará todos los fuentes en binarios ejecutables.

La instalación instalará los binarios y cualquier archivo de soporte en el directorio apropiado.

El comando `mkdir` creará un nuevo directorio llamado squid bajo */var/log*.

El comando `rm -rf` removerá el directorio */var/logs* desde este directorio se manejará los archivos log relativos al *Squid* que serán movidos a el directorio */var/log/squid*.

El comando `chown` cambiará el dueño del directorio */var/log/squid* al usuario *squid*.

El comando `chmod` cambiará los permisos de los directorios de caché por obvias razones de seguridad.

Cuando nosotros removemos los pequeños scripts llamados *RunCache* y *RunAccel* cuando iniciamos el *Squid* pasamos a modo de cacheo ó modo acelerado, desde que nosotros usamos un mejor script localizado bajo el directorio */etc/rc.d/init.d/* que toma las ventajas de UNIX system V.

El comando `strip` reducirá el tamaño de los binarios para mayor eficiencia.

Luego removemos los directorios fuentes que no usaremos:

```
[root@lhost] /# cd /var/tmp
[root@lhost ]/tmp# rm -rf squid-version/ squid-version.STABLEz-src.tar.gz
[root@lhost ]/tmp# rm -rf malloc/ malloc.tar.gz (if you used the GNU malloc external library).
```

6.4 CONFIGURACION DEL SQUID.

Para ejecutar el servidor *Squid* en modo *httpd-acelerado*, los siguientes archivos son requeridos, que deben de ser creados ó copiados en los directorios apropiados:

- Archivo squid.conf en el directorio /etc/squid/.
- Archivo script squid en el directorio /etc/rc.d/init.d/.
- Archivo squid en el directorio /etc/logrotate.d/.

Para ejecutar el servidor *Squid* en modo proxy-cacheo, los siguientes archivos son requeridos en los directorios apropiados:

- Archivo squid.conf en el directorio /etc/squid/.

- Archivo script squid script en el directorio /etc/rc.d/init.d/.

- Archivo squid en el directorio /etc/logrotate.d/.

6.4.1 Configurando el archivo /etc/squid/squid.conf en modo httpd-accelerator. El archivo *squid.conf* es usado para configurar las diferentes opciones de nuestro servidor proxy *Squid*.

En el archivo de configuración nosotros configuramos este archivo en modo *httpd-acelerado*. En este modo acelerado, si el servidor web se ejecuta en el mismo servidor donde el *Squid* se encuentra instalado, podemos colocar ese demonio en el puerto 81.

Con el servidor *Apache*, podemos asignarle el puerto 81 al servicio *httpd* desde el archivo *httpd.conf*. Sí el servidor necesita usar dicho puerto podemos colocar al *Squid* en un número IP donde el puerto 80 no esté en uso. Editamos el archivo *squid.conf* y cambiamos las siguientes opciones:

```
http_port 80
#Esta poción especifica el número del puerto donde el Squid
#escucha las peticiones de los clientes. Sí la opción es 80
#los clientes tendrán la ilusión de estar conectados al
#servidor Apache. Cuando el Squid se encuentre en modo
#acelerado debe estar en ese puerto.
icp_port 0
#Esta opción especifica el número del puerto donde Squid
#envía y recibe las peticiones ICP de los caches vecinos. Al
#estar en 0 se deshabilita. Esta característica solo es
#necesaria para un cacheo de multi-nivel.
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
#Estas dos opciones son usadas para forzar que ciertos
#objetos no sean ubicados en la caché, estos archivos se
#encuentran bajo el directorio cgi-bin. Esta es una opción de
#seguridad.
cache_mem 16 MB
#Esta opción especifica la memoria RAM que puede usar el
#caché al ser llamado. Alguno de los objetos In-Transit, Hot,
#Negative-Cached. Así pues si tenemos 48 MB libres para Squid
```

```
#será 48/3 = 16, la memoria asignada para cada uno de los
#objetos.
cache_dir ufs /cache 200 16 256
#Esta opción especifica en orden que tipo de sistema de
#alocación se usa, ufs; el nombre del directorio de caché,
#/cache; el espacio en disco a usar, 200MB; el número del
#primer nivel de subdirectorios a crear bajo el directorio de
#caché, 16; y el número del segundo nivel de subdirectorios a
#crear bajo el directorio de caché, 256.
emulate_httpd_log on
#Al tener activa esta opción quiere decir que el Squid puede
#emular los archivos del log del servidor Apache.
redirect_rewrites_host_header off.173
#Esto quiere decir que el Squid “no” rescribirá en las
#cabeceras de las peticiones del servidor.
replacement_policy GDSF
#Esta opción es usada para eliminar objetos que no se usen
#para hacer mas espacio que se puedan necesitar para el
#cacheo. Esta opción solo es disponible si se configura desde
#la instalación.
acl all src 0.0.0.0/0.0.0.0
```

```
http_access allow all
#Estas opciones especifican la lista de control de acceso que
#pueda tener, se accede a todo sí usamos el modo acelerado.
cache_mgr admin@sistema.cutb.edu.co
#Esta opción especifica el mail del administrador del
#sistema.
cache_effective_user squid
cache_effective_group squid
#Estas dos opciones especifican los identificadores de
#usuarios y grupos que pueden ejecutar Squid.
httpd_accel_host 208.164.186.3
httpd_accel_port 80
#Estas dos opciones recogen el IP del servidor Apache para
#dar la ilusión de conexión.
log_icp_queries off
#Con esta opción especificamos los queries que necesitamos
#para rastrear los caché vecinos, el log de los queries se
#almacenará en el archivo access.log ó no de acuerdo a lo que
#se quiera. Para modo acelerado colocamos OFF.
cachemgr_passwd my-secret-pass all
# Esta opción especifica la contraseña que puede ser
```

```
#requerida para el acceso a las operaciones de la utilidad de
#programa cgi cachemgr.cgi. Esta utilidad CGI es designada
#para brindar estadísticas sobre la configuración del Squid y
#el desempeño que este tenga. En el campo my-secret-pass se
#especifica la contraseña para darle opción a este programa.
buffered_logs on
#Esta opción es puesta en ON cuando se quiere brindar
#velocidad a la escritura de algunos archivos de log.
```

6.4.2 Configurando el archivo /etc/squid/squid.conf en modo proxy-cacheo.

Con las menores modificaciones del archivo *squid.conf* que nosotros tenemos definimos del modo acelerado, nosotros podemos ejecutar *Squid* como modo proxy-cacheo, para todos los usuarios en la red que usaremos *Squid* para acceder a Internet.

Con esta configuración, podemos completar el control, y aplicar políticas especiales las cuales podremos ver, acceder y descargar. Podemos. controlar el ancho de banda usado, tiempo de conexión, entre otros.

Un servidor proxy cache puede ser confgurado para ejecutarse como servidor sólo para una corporación, ó puede usarse para la administración de caches heredando de otros servidores proxy alrededor de la Internet.

Con el primer ejemplo configuraremos *Squid* como un servidor sólo, y hablaremos un poco de la configuración heredada, donde dos ó más servidores proxy-caché cooperarán para servirse documentos para cada orden.

Editamos el archivo *squid.conf* y cambiamos/añadimos los siguientes comandos para un servidor proxy sólo (ermitaño#):

```
http_port 8080
icp_port 0
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 16 MB
cache_dir ufs /cache 200 16 256
redirect_rewrites_host_header off
replacement_policy GDSF
acl localnet src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
```

```
acl Safe_ports port 80 443 210 119 70 21 1025-65535
acl CONNECT method CONNECT
acl all src 0.0.0.0/0.0.0.0
http_access allow localnet
http_access allow localhost
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny all
cache_mgr admin@openna.com
cache_effective_user squid
cache_effective_group squid
log_icp_queries off
cachemgr_passwd my-secret-pass all
buffered_logs on
```

La diferencia entre este modo y el acelerado es el uso de las listas de control de accesos (*ACL*). Esta opción nos sirve para restringir el acceso basado en la dirección IP fuente (*src*), la dirección IP del destino (*dst*), el dominio fuente, el dominio de destinación, tiempo y otras. Muchos tipos existen con esta opción y solo consultamos el archivo *squid.conf* para una lista completa. Los cuatros más usados son:

```
acl  nombre   tipo     datos
 |     |        |        |
acl some-name src a.b.c.d/e.f.g.h
# ACL restrict access based on source IP address
acl some-name dst a.b.c.d/e.f.g.h
# ACL restrict access based on destination IP address.175
acl some-name srcdomain foo.com
# ACL restrict access based on source domain
acl some-name dstdomain foo.com
# ACL restrict access based on destination domain
```

Como un ejemplo para restrngir el acceso a nuestro servidor proxy Squid, el uso sea solo para clientes inernos y especificar un rango de puertos designados, integramos las siguientes líneas:

```
acl localnet src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl Safe_ports port 80 443 210 119 70 21 1025-65535
acl CONNECT method CONNECT
acl all src 0.0.0.0/0.0.0.0
http_access allow localnet
```

```
http_access allow localhost
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny all
```

Esta configuración *acl* permitira a los clientes de los distintos IP privados acceder al servidor proxy; es recomendado que tenga el IP local de defecto para acceder al proxy.

Después cambiaremos el rango de los puertos (80=http, 443=https, 210=wais, 119=nntp, 70=gopher, y 21=ftp) para que nuestros clientes internos puedan acceder al Internet, nosotros negaremos el método de conexión para prevenir que usuarios externos intenten conectarse con el servidor proxy y finalmente, negaremos toda dirección IP y puerto en nuestro servidor proxy.

Cacheo de multi-nivel, el segundo tipo de servidor proxy cache es el de multi nivel donde varios servidores cooperan para compartir documentos. Con este método, la organización usa el caché de muchos servidores caché y compensa aceptando a otros servidores de caché usen el suyo. Es importante notar que en esta situación, el proxy caché puede jugar dos roles diferentes en la herencia. Este puede configurarse para compartir solamente documentos, ó compartir

información de determinada ubicación. Una buena estrategia es generando más trafico de la red sin un cacheo de web eligiendo solamente un pequeño numero de servidores cache.

6.4.3 Configurando el archivo script /etc/rc.d/init.d/squid/ para todos los modos. El archivo sirve para inicializar ó terminar este servicio. Este script ha sido modificado para configurar un intercambio de caché usando el directorio */var/spool/squid* del *squid*.

Creamos el archivo script con el comando, `touch /etc/rc.d/init.d/squid` y añadimos:

```
#!/bin/bash.176
# Este script inicia ó detiene el servicio Squid
#Squid Internet Object Cache
#
# chkconfig: - 90 25
# pidfile: /var/run/squid.pid
# config: /etc/squid/squid.conf
PATH=/usr/bin:/sbin:/bin:/usr/sbin
```

```
export PATH # Fuente de las librerías de funciones.
. /etc/rc.d/init.d/functions
# Configuración de la fuente de la red.
. /etc/sysconfig/network # Verifica que la red esté activa.
[ ${NETWORKING} = "no" ] && exit 0
# verifica si squid.conf está
[ -f /etc/squid/squid.conf ] || exit 0
#ubique el binario squid
[ -f /usr/sbin/squid ] && SQUID=squid
[ -z "$SQUID" ] && exit 0
# determine el directorio de intercambio
CACHE_SWAP=`sed -e 's/#.*//g' /etc/squid/squid.conf | \
grep cache_dir | sed -e 's/cache_dir//' | \
cut -d ' ' -f 2`
[ -z "$CACHE_SWAP" ] && CACHE_SWAP=/cache
#opciones por defecto
# -D deshabilita la verificación de DNS. Si muestra que no
# hay conexión a Internet
# cuando comienza el squid no comente esto
#SQUID_OPTS="-D" RETVAL=0
case "$1" in
```

```
start)
echo -n "Starting $SQUID: "
for adir in $CACHE_SWAP; do
if [ ! -d $adir/00 ]; then
echo -n "init_cache_dir $adir... "
$SQUID -z -F 2>/dev/null
fi
done
$SQUID $SQUID_OPTS &
RETVAL=$?
echo $SQUID
[ $RETVAL -eq 0 ] && touch /var/lock/subsys/$SQUID
;; stop)
echo -n "Stopping $SQUID: "
$SQUID -k shutdown &
RETVAL=$?
if [ $RETVAL -eq 0 ] ; then
rm -f /var/lock/subsys/$SQUID
while : ; do
[ -f /var/run/squid.pid ] || break
sleep 2 && echo -n "."
```

```
done
echo "done"
else
echo.177
fi
;; reload)
$SQUID $SQUID_OPTS -k reconfigure
exit $?
;; restart)
$0 stop
$0 start
;; status)
status $SQUID
$SQUID -k check
exit $?
;; probe)
exit 0;
;; *)
echo "Usage: $0 {start|stop|status|reload|restart}"
exit 1
esac exit $RETVAL
```

Ahora hacemos el script ejecutable y cambiamos los permisos por defecto:

```
[root@lhost /]# chmod 700 /etc/rc.d/init.d/squid
```

Creamos el vinculo simbólico para squid con el comando:

```
[root@lhost /]# chkconfig --add squid
```

Por defecto el script squid no inicializará el servicio al reiniciar el sistema por ende si queremos que se efectué esta opción debemos de ejecutar el siguiente comando:

```
[root@lhost /]# chkconfig --level 345 squid on
```

Iniciamos nuestro servidor proxy manualmente de la siguiente manera:

```
[root@lhost /]# /etc/rc.d/init.d/squid start
Starting squid: init_cache_dir ufs... squid
```

6.4.4 Configurando el archivo /etc/logoratate.d/squid. De esta manera configuramos la rotación de los archivos log cada semana.

Creamos el archivo de la forma touch /etc/logrotate.d/squid y añadimos:

```
/var/log/squid/access.log {
weekly
rotate 5
copytruncate
compress
notifempty
missingok.178
}
/var/log/squid/cache.log {
weekly
rotate 5
copytruncate
compress
notifempty
missingok
} /var/log/squid/store.log {
```

```
weekly
rotate 5
copytruncate
compress
notifempty
missingok
# Este script le pregunta al squid cada cuanto rotara el
archivo de log.
# Reinicia el squid en un proceso largo y solo deja activo el
log.
postrotate
/usr/sbin/squid -k rotate
endscript
}
```

6.4.5 Asegurando e inmunizando el Squid. Para tener más control del directorio de caché montado, podemos agregarle propiedades a esta partición para mejorar la seguridad del caché. Asumamos que *dev/sda8* es la partición en el sistema donde el directorio */cache* de *Squid*, debemos editar el archivo *fstab* y cambiar la línea:

```
/dev/sda8/cacheext2defaults 1 2
```

Por:

```
/dev/sda8/cache ext2noexec,nodev,nosuid 1 2
```

No olvidemos remontar la partición para que los cambios surjan efectos. A la vez podemos inmunizar el archivo de configuración squid.conf, de la siguiente manera para evitar cambios inesperados:

```
[root@lhost /]# chattr +i /etc/squid/squid.conf
```

6.5 OPTIMIZANDO EL SQUID.

Los atributos atime y noatime pueden ser usados para obtener un desempeño considerado ganado en el directorio de caché. Los recursos más importantes para Squid es la memoria física. Su procesador no necesita ser ultra-rápido. El disco de sistema debe ser tan veloz como al volumen de información en el caché. No use discos *IDE* si quiere ayudarse, a menos que sean *ULTRA-ATA*.

6.5.1 La utilidad cachemgr.cgi. La utilidad *cachemgr.cgi*, es disponible desde el momento de compilar e instalar *Squid* en nuestro sistema y es diseñada para manejar una interfaz web, ofreciendo estadísticas sobre la configuración y desempeño del *Squid*.

Este programa esta localizado en el directorio */usr/lib/squid* y puede colocarlo en su directorio cgi-bin de su servidor web. Los siguientes son pasos para usar este programa.

Movamos esta utilidad cgi a nuestro directorio cgi-bin por defecto:

```
[root@lhost /]# mv /usr/lib/squid/cachemgr.cgi
/home/httpd/cgi-bin
```

Asumiendo este directorio por ser el defecto, este puede cambiar de acuerdo a la configuración del sistema. Cuando coloque este programa, puede observarlo desde su browser a la siguiente dirección http://my-web-server/cgi-bin/cachemgr.cgi. Donde my-web-server es la dirección del servidor web.

Si tiene configurado el archivo *squid.conf* para usar contraseñas de autenticación para *cachemgr.cgi*, deberá preguntarse para entrar a la información de servidor de

caché, puerto de caché, nombre del administrador del servicio, y contraseña antes de acceder al programa *cachemgr.cgi*.

Cuando sea autenticado por el servidor, deberá ver en la interfaz del menú del manejador de caché donde podemos examinar y analizar las diferentes opciones relativas a nuestro servidor proxy *Squid*.

6.6 ARCHIVOS INSTALADOS

Los archivos instalados por el programa *Squid* son:

```
/etc/squid
/etc/squid/mib.txt
/etc/squid/squid.conf.default
/etc/squid/squid.conf
/etc/squid/mime.conf.default
/etc/squid/mime.conf
/etc/squid/errors.180
/etc/squid/errors/ERR_ACCESS_DENIED
/etc/squid/errors/ERR_CACHE_ACCESS_DENIED
```

/etc/squid/errors/ERR_CACHE_MGR_ACCESS_DENIED

/etc/squid/errors/ERR_CANNOT_FORWARD

/etc/squid/errors/ERR_CONNECT_FAIL

/etc/squid/errors/ERR_DNS_FAIL

/etc/squid/errors/ERR_FORWARDING_DENIED

/etc/squid/errors/ERR_FTP_DISABLED

/etc/squid/errors/ERR_FTP_FAILURE

/etc/squid/errors/ERR_FTP_FORBIDDEN

/etc/squid/errors/ERR_FTP_NOT_FOUND

/etc/squid/errors/ERR_FTP_PUT_CREATED

/etc/squid/errors/ERR_FTP_PUT_ERROR

/etc/squid/errors/ERR_FTP_PUT_MODIFIED

/etc/squid/errors/ERR_FTP_UNAVAILABLE

/etc/squid/errors/ERR_INVALID_REQ

/etc/squid/errors/ERR_INVALID_URL

/etc/squid/errors/ERR_LIFETIME_EXP

/etc/squid/errors/ERR_NO_RELAY

/etc/squid/errors/ERR_ONLY_IF_CACHED_MISS

/etc/squid/errors/ERR_READ_ERROR

etc/squid/errors/ERR_READ_TIMEOUT

/etc/squid/errors/ERR_SHUTTING_DOWN

```
/etc/squid/errors/ERR_SOCKET_FAILURE
/etc/squid/errors/ERR_TOO_BIG
/etc/squid/errors/ERR_UNSUP_REQ
/etc/squid/errors/ERR_URN_RESOLVE
/etc/squid/errors/ERR_WRITE_ERROR
/etc/squid/errors/ERR_ZERO_SIZE_OBJECT
/etc/rc.d/init.d/squid
/etc/rc.d/rc0.d/K25squid
/etc/rc.d/rc1.d/K25squid
/etc/rc.d/rc2.d/K25squid
/etc/rc.d/rc3.d/S90squid
/etc/rc.d/rc4.d/S90squid
/etc/rc.d/rc5.d/S90squid
/etc/rc.d/rc6.d/K25squid
/etc/logrotate.d/squid
/usr/lib/squid
/usr/lib/squid/dnsserver
/usr/lib/squid/unlinkd
/usr/lib/squid/cachemgr.cgi
/usr/lib/squid/icons
/usr/lib/squid/icons/anthony-binhex.gif
```

```
/usr/lib/squid/icons/anthony-bomb.gif
/usr/lib/squid/icons/anthony-box.gif
/usr/lib/squid/icons/anthony-box2.gif.181
/usr/lib/squid/icons/anthony-c.gif
/usr/lib/squid/icons/anthony-compressed.gif
/usr/lib/squid/icons/anthony-dir.gif
/usr/lib/squid/icons/anthony-dirup.gif
/usr/lib/squid/icons/anthony-dvi.gif
/usr/lib/squid/icons/anthony-f.gif
/usr/lib/squid/icons/anthony-image.gif
/usr/lib/squid/icons/anthony-image2.gif
/usr/lib/squid/icons/anthony-layout.gif
/usr/lib/squid/icons/anthony-link.gif
/usr/lib/squid/icons/anthony-movie.gif
/usr/lib/squid/icons/anthony-pdf.gif
/usr/lib/squid/icons/anthony-portal.gif
/usr/lib/squid/icons/anthony-ps.gif
/usr/lib/squid/icons/anthony-quill.gif
/usr/lib/squid/icons/anthony-script.gif
/usr/lib/squid/icons/anthony-sound.gif
/usr/lib/squid/icons/anthony-tar.gif
```

```
/usr/lib/squid/icons/anthony-tex.gif
/usr/lib/squid/icons/anthony-text.gif
/usr/lib/squid/icons/anthony-unknown.gif
/usr/lib/squid/icons/anthony-xbm.gif
/usr/lib/squid/icons/anthony-xpm.gif
/usr/sbin/RunCache
/usr/sbin/RunAccel
/usr/sbin/squid
/usr/sbin/client
/var/log/squid
```

7 CONCLUSIONES

Al terminar el proyecto pude sacar las siguientes conclusiones:

- El diseño de cualquier tipo de servicio se ve limitado al uso y usuarios que dispondrá.

- En los sistemas operativos multiusuarios la seguridad viene representada por el administrador y es él el responsable de este item.

- Todos los servicios TCP/IP no son compatibles entre sí, sino por el contrario muchas de las veces presentan problemas al convivir dentro de una máquina.

- Los servicios TCP/IP pueden ser distribuidos, es decir que cada máquina pueda ejecutar un servicio, pero todas ellas pueden formar un conjunto es decir que puede representarse como una sola.

- La configuración de los servicios TCP/IP es siempre un archivo de texto.

- Toda instalación debe estar preparada por un diseño, la instalación debe llevar un informe antes de la instalación para poder llevar un orden en los estatutos del servidor.

- El administrador debe ser una persona preparada en el campo de diseño, análisis y técnico ya que cada una de ellas es necesaria para cada una de ella, el diseño proviene del análisis y de la parte técnica, y así entre todos estos aspectos.

- Las auditorias a hacer para los servicios ofrecidos deben llevar un informe acerca de la utilidad, desempeño y uso del servicio, de modo que el administrador pueda saber que actualizar ó cambiar en la red.

- Administrar e instalar servicios en una máquina es muy sencillo, al entrar al punto de la configuración personal, que permite encontrar ese hueco que puede llegar a ser riesgo para todo nuestro montaje, ese punto de vista se va complicando, lo mejor es configurar e instalar cada uno de los servicios manualmente.

- Las casas distribuidoras ofrecen poca información al tratarse del software que ofrece, por ende es tarea de uno buscar ayuda entre todos los usuarios para la resolución de problemas.

- Para el diseño e instalación de servicios TCP/IP en Linux no se tiene que ser un gurú en la programación, ni un hacker ó algo por el estilo, pero si desea hacer algo bien hecho, “falta mucho por aprender”; puesto que cada servicio actualmente ofrece una ilimitada lista de opciones que nos beneficiarán a la hora de ofrecer el servicio, por ende el administrador debe delegar especialidades a varios administradores en caso de una red extensa para que cada uno se encarge del servicio que tenga asociado. Un administrador para una gran cantidad de servicios, que tenga seguridad de alto nivel ó gran volumen de información, no daría el alcance apropiado para el manejo de la red.

- La separación de servicios puede darse de acuerdo a las políticas de la empresa, seguridad del sistema, cantidad de usuarios ó volumen de información en tráfico.

- Todo administrador debe contar con un poco de sentido común, puesto que el diseño de un servicio es independiente para cada caso.

BIBLIOGRAFÍA

TACKETT, Jack y BURNETT, Steve. Edición especial LINUX 4ta edición. Madrid: PEARSON Educación S.A., 2000. 1110 p.

MOURANI, Gerhard. Securing and optimizing LINUX: Red Hat Edition. New York: Open Docs, 2000. 486 p.

ANONIMO. Edición especial LINUX. Máxima Seguridad. Madrid: PEARSON Educación S.A., 2000. 808 p.

Java Magazine. Año 2 Número 3 (ago-sept. 2000).Madrid: Prensa técnica, 2000. 84 p. Mensual.

Solo LINUX. Año 1 Número 1 (mar-abr. 1999). Madrid: Prensa técnica, 1999. 84 p. Mensual.

Programación Actual. Año 1 Número 2 (may-jun. 1997). Madrid: Prensa técnica, 1997. 84 p. Mensual.

Programación Actual. Año 1 Número 3 (jun-jul. 1997). Madrid: Prensa técnica, 1997. 84 p. Mensual.

Programación Actual. Año 1 Número 5 (ago-sep. 1997). Madrid: Prensa técnica, 1997. 84 p. Mensual.

Programación Actual. Año 1 Número 7 (oct-nov. 1997). Madrid: Prensa técnica, 1997. 84 p. Mensual.

Programación Actual. Año 1 Número 8 (nov-dic. 1997). Madrid: Prensa técnica, 1997. 84 p. Mensual.

Programación Actual. Año 1 Número 11 (abr-may. 1998). Madrid: Prensa técnica, 1998. 84 p. Mensual.

Programación Actual. Año 1 Número 12 (may-jun. 1998). Madrid: Prensa técnica, 1998. 84 p. Mensual.

Programación Actual. Año 2 Número 13 (jun-jul 1998). Madrid: Prensa técnica, 1998. 84 p. Mensual..185

Programación Actual. Año 2 Número 14 (jul-ago. 1998). Madrid: Prensa técnica, 1998. 84 p. Mensual.

ANEXO A. LA LICENCIA GNU.

Copyright (C) 1989, 199 Printed below is the GNU General Public License (the *GPL* or *copyleft*), under which Linux is licensed. It is reproduced here to clear up some of the confusion about Linux's copyright status--Linux is *not* shareware, and it is *not* in the public domain. The bulk of the Linux kernel is copyright ©1993 by Linus Torvalds, and other software and parts of the kernel are copyrighted by their authors. Thus, Linux *is* copyrighted, however, you may redistribute it under the terms of the GPL printed below.

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

PREÁMBULO

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software-to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to

certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent

licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMINOS Y CONDICIONES.

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program'', below, refers to any such program or work, and a “work based on the Program'' means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term ``modification''.) Each licensee is addressed as “you''.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not

restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program. You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.).

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License,

whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted

only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and ”any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free

software and of promoting the sharing and reuse of software generally. NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM ``AS IS'' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT

LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

COMO APLICA A LOS NUEVOS PROGRAMAS

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty;

and each file should have at least the ``copyright'' line and a pointer to where the full notice is found.

One line to give the program's name and a brief idea of what it does.

Copyright ©19yy name of author This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type ‘show w'. This is free software, and you are welcome to redistribute it under certain conditions; type ‘show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items-whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program ‘Gnomovision' (which makes passes at compilers) written by James Hacker.
signature of Ty Coon, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

ANEXO B. LA LICENCIA GPL.

EL PRESENTE TEXTO ES UNA TRADUCCION A PROPOSITO EXCLUSIVAMENTE INFORMATIVO DE LOS TERMINOS DE LA LICENCIA DE LINUX-MANDRAKE. EN NINGUN CASO LA PRESENTE TRADUCCION TIENE VALOR LEGAL SIENDO OFICIAL EXCLUSIVAMENTE LA VERSION EN FRANCES DE LA LICENCIA DE LINUX-MANDRAKE.

No obstante, esperamos que esta traducción ayudará a los que hablan Castellano a entenderla mejor.

Introducción

El conjunto de elementos que incluye el sistema operativo y los diferentes componentes disponibles en la distribución Mandrake Linux son nombrados en adelante "Programas". Los programas incluyen en particular, pero de manera no limitativa, el conjunto de programas, procedimientos, reglas y documentaciones relativas al sistema operativo y a los diferentes componentes de la distribución

Mandrake Linux.

1. Licencia

Le rogamos leer cuidadosamente este documento. Éste constituye un contrato de licencia entre Ud. (persona física o persona moral) y MandrakeSoft S.A. aplicado a los programas. El hecho de instalar, de reproducir o de usar los programas de cualquiera manera que sea indica que Ud. reconoce haber tenido conocimiento preliminar, y aceptado conformarse a los términos y condiciones del presente contrato de licencia. En caso de desacuerdo con el documento presente no esta autorizado a instalar, reproducir y usar de cualquiera manera que sea este producto. El contrato de licencia sera automáticamente anulado sin aviso previo en el caso que no se conformaría a las disposiciones de este documento. En caso de anulación Ud. tendrá que anular inmediatamente todo ejemplar y todas las copias de todos los programas y de todas las documentaciones que constituyen el sistema operativo y los diferentes componentes disponibles en en la distribución Mandrake Linux.

2. Garantía y limitaciones de garantía

Los programas y la documentación que los acompaña son proporcionados en el estado y sin garantía ninguna. MandrakeSoft S.A. déclina toda no se responsabiliza de las consecuencias de un daño directo, especial, indirecto o accesorio, de cualquiera naturaleza que sea, en relación con la utilización de los programas, en particular y de manera no limitativa, todos daños resultados por perdidas de beneficio, interrupción de actividad, pérdida de informaciones Comerciales u otras perdidas financieras, así que por eventuales condenaciones y indemnizaciónes debidas consecuentes a una decisión de justicia, y eso incluso si MandrakeSoft S.A. fue informada de la aparición o eventualidad de tales daños.

ADVERTENCIA EN CUANTO AL POSEER O USO DE PROGRAMAS PROHIBIDOS EN CIERTOS PAÍSES

En ningún caso, ni MandrakeSoft S.A. ni sus proveedores podrán ser tenidos responsables por un perjuicio especial, directo, indirecto o accesorio, de cualquiera naturaleza que sea (en particular y de manera no limitativa, perdidas de beneficio, interrupción de actividad, pérdida de informaciones comerciales u otras perdidas financieras, así que por eventuales condenaciones y indemnizaciónes debidas consecuentes a una decisión de justicia) que resultaría de la utilización, detención o simple baja desde une de los sitios de transferencia de Mandrake

Linux de programas prohibidos por la legislación a la cual Ud. esta sometido. Esta advertencia se aplica en particular a algunos programas de criptografia suplidos con los programas.

3. Licencia GPL y otras licencias

Los Programas están constituidos por módulos de programas creados por diversas personas (físicas o morales). Muchos de ellos están distribuidos bajo los términos de la Licencia Pública General GNU (denominada aquí abajo "GPL") u otras licencias parecidas. La mayoría de estas licencias le permiten copiar, adaptar o redistribuir los módulos de programas que cubren. Favor leer y aceptar los términos y condiciones de las licencias acompañando cada uno de ellos antes de usarlos. Toda pregunta relativa a la licencia se debe someter al autor (o su representante) del dicho programa, y no a MandrakeSoft. Los programas desarrollados por MandrakeSoft son sometidos a la licencia GPL. La documentación escrita por MandrakeSoft esta sometida a una licencia especifica. Favor de referirse a la documentación para obtener más información.

4. Propiedad intelectual

Todos los derechos, títulos e intereses de los diferentes Programas son la propiedad exclusiva de sus autores respectivos y son protegidos por el derecho de propiedad intelectual y otras leyes aplicadas al derecho de programas. Las marcas "Mandrake" y "Mandrake Linux" así como los logótipos asociados son registrados por MandrakeSoft S.A.

5. Disposiciones diversas

Si alguna disposición de este contrato de licencia estuviera declarada nula, ilegal o inaplicable por un tribunal competente, esta disposición sera excluida del presente contrato. Sin embargo Ud. seguirá sometido a las otras disposiciones, que recibirán sus plenos efectos. El contrato de licencia es sometido a la Ley francesa. Toda pregunta o protesta relativa a las presentes sera resuelto en primer lugar por vía amigable. En caso de desacuerdo con MandrakeSoft S.A., el litigio será sometido a los tribunales competentes de París, Francia. Para toda pregunta relacionada con este documento, favor dirigirse a MandrakeSoft S.A.

ANEXO C. LICENCIA DEL APACHE

/*

==========

=====

* Licencia del programa Apache, Version 1.1

*

* Copyright (c) 2000 The Apache Software Foundation. Todos los derechos

* reservados.

*

* Redistribucion en usos binarios y fuentes, con ó sin modificación,

* son permitidas siguiendo y conociendo las siguientes condiciones:

*

*

* 1. Las redistribuciones del código fuente pueden ser transferidas

* con las noticias de los derechos de copia, esta lista de condiciones y

* y la siguiente aclaración.

*

* 2. Las redistribuciones pueden ser transferidas con la noticia de los

* derechos de copia, esta lista de condiciones y la siguiente aclaración

* en la documentación y/u otros materiales providas en la distribución.

*

* 3. La documentación del usuario final incluida con la redistribución

* cualquiera, debe colocar el siguiente texto

* "This product includes software developed by the

* Apache Software Foundation (http://www.apache.org/)."

* Alternamente, este texto puede aparecer en el software mismo,

* siempre y cuando el texto aparezca sin cambios.

*

* 4. Los nombres "Apache" y "Apache Software Foundation" no deben

* ser usados para promover productos derivados de este software

* sin la autorización previa por escrito de los permisos.

* Para escribir los permisos, por favor contáctenos a apache@apache.

*

* 5. Los productos derivados de este software no pueden llamarse "Apache"

* Apache no puede aparecer en su nombre, sin la autorización previa

* escrita en los permisos de la fundación de software Apache.

*

* ESTE SOFTWARE ES PROVEIDO COMO ES Y CUALQUIER IMPLICACION

O GARANTIAS

* INCLUYENDO, LIMITADO A, LAS GARANTIAS DE MERCADO Y USO PARA UN PROPOSITO
* PARTICULAR SON RECHAZADOS. EN NINGUN EVENTO DEBE APARECER LA FUNDACION DE
* SOFTWARE APACHE O SUS CONTRIBUDORES POR ACCIONES DIRECTAS, INDIRECTAS
* INCIDENCIALES, ESPECIALES, EJEMPLARES, O POR CONSECUENCIAS DE DAÑOS
* (INCLUYENDO LA NO LIMITACION, PERDIDA DE SU USO, PERDIDA DE DATOS,
* INTERRUPCION EN LOS NEGOCIOS)CUALQUIER CAUSA O TEORIA DE LEY, CUALQUIER
* CONTRATO, LEY ESTRICTIVA, O NEGLIGENCIA EVADIMOS CUALQUIER RESPONSABILIDAD
* ANTE LA ADVERTENCIA QUE ESTE SOFTWARE PUEDE OCASIONAR CUALQUIER TIPO DE
* DAÑO.
*
==========

*

* Este software consiste de contribuciones voluntarias hechas por muchos

* individuos benefactores de la Fundación de Software Apache. Para más

* información de la Fundación de Software Apache, por favor visite

* <http://www.apache.org/>.

*

* porciones de este software estan basados en un software de uso público

* originalmente escrito en el Centro Nacional para Aplicaiones de Supercómputos,

* Universidad de Illinois, Urbana-Champaign.

*/

ANEXO D. LISTADO DE RFC COMÚNMENTE USADOS

A continuación hay una lista de los RFCs mencionados a lo largo del libro. Todos los RFCs están disponibles vía FTP anónimo en los servidores nic.ddn.mil, ftp.uu.net. Para obtener un RFC vía correo electrónico, envíe un mensaje a service@nic.ddn.mil, incluyendo la petición send RFC-*numer*o.TXT en el *asunto.*

1340

Números Asignados, *Postel, J*., y *Reynolds, J.* El RFC de Números Asignados define el significado de los números empleados por los diferentes protocolos, tales como los números de puertos estándar atendidos por los servidores TCP y UDP, así como los números de protocolo contenidos en la cabecera del datagrama IP.

1144

Compresión de cabeceras TCP/IP para enlaces tipo serie de baja velocidad, *Jacobson, V.* Este documento describe el algoritmo usado para comprimir las cabeceras TCP/IP en CSLIP y PPP. Realmente merece la pena leerlo!

1033

Guía de los Administradores de Dominio, *Lottor, M.* Junto con los RFCs complementarios, RFC 1034 y RFC 1035, es la fuente de información más completa sobre DNS, Sistema de resolución de nombres de dominio.

1034

Nombres de Dominios - Conceptos y Características *Mockapetris, P.V.* Complemento de RFC 1033.

1035

Nombres de Dominios - Implementación y Especificación, *Mockapetris, P.V.* Complemento de RFC 1033.

974

Distribución de Correo Electrónico y el Sistema de Dominios, *Partridge, C.* Este RFC describe la distribución de correo en Internet. Lea esto para conocer la historia completa de los registros MX ...

977

Protocolo de Transferencia de Noticas de Red, *Kantor, B.*, y *Lapsley, P.* Definición del NNTP, el método más común de transporte de noticias de red en Internet.

1094

NFS: Especificación del Protocolo de Sistema de Fichero de Red *Nowicki, B.* Especificación formal de NFS y del protocolo de montaje (version 2).

1055

No-standard de Transmisión de Datagramas de IP sobre líneas de Comunicación Serie: SLIP, *Romkey, J.L.* Describe SLIP, el Protocolo de Comunicación via línea Serie de Internet.

1057

RPC: Especificación del Protocolo de Llamada a Procedimiento Remoto: Version 2, *Sun Microsystems, Inc*

1058

Protocolo de Información de Rutas, *Hedrick, C.L.* Describe RIP, protocolo usado para el intercambio de información de rutas en LANs y MANs.

821

Protocolo Simple de Transferencia de Correo,*Postel, J.B.* Define SMTP, el protocolo de transporte de correo sobre TCP/IP.

1036

Estándar para el Intercambio de mensajes USENET, *Adams, R.* y *Horton, M.R.* Este RFC describe el formato de los mensajes de noticias USENET, y cómo son intercambiados tanto en Internet como en redes UUCP. Se espera que pronto haya una revisión de este protocolo.

822

Estándar sobre el Formato de mensajes de texto en Internet de ARPA, *Crocker, D.* Es la más completa fuente de sabiduría sobre el correo que cumple el estándar RFC. Todo el mundo lo conoce, aunque son pocos los que realmente lo han leído.

968

Twas The Night Before Startup, *Cerf, V.* Quién dice que los heroes de las redes no son poetas?. (El título no se tradujo por ser obra poética).