ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

"DESARROLLO DE UNA GUIA REFERENCIAL PARA EL ANÁLISIS Y OPTIMIZACIÓN TECNOLÓGICA DE SERVIDORES WEB APLICADO A LA ESPOCH".

TESIS DE GRADO

Previa la obtención del título de:

INGENIEROS EN SISTEMAS

Presentado por:

MARITZA GUADALUPE ORELLANA BAJAÑA

JUAN CARLOS BAJAÑA HELENO

RIOBAMBA - ECUADOR

- 2007 -

Agradecemos a Dios por habernos dado la vida y la guia para poder lograr esta meta en nuestras vidas.
Agradecemos al Ing. Danilo Pastor Director de la Tesis, al Ing. Wladimir Castro y a la Ing. Landy Ruiz Miembros de la misma, por la ayuda y cooperación que nos brindaron en la ejecución de este trabajo. Asi mismo al Ing Paul Bernal por toda la cooperación que nos dio para la realización del mismo

Juan y Maritza

Agradezco a mis padres Luis y Cresencia por haberme apoyado en todo momento, por haberme dado las fuerzas necesarias en los momentos en que ya no las tenía. Agradezco a mis hermanos Ruth, Luis, Leonor, Hermel, David y Carolina por haberme apoyado y ayudado en los momentos en que mas lo necesitaba. Agradezco a mis amigos y compañeros ya que sin su ayuda no habria podido concluir este trabajo

Maritza

Le agradezco de todo corazón a mi madre por el apoyo brindado para poder hacer realidad mi sueño de culminar mi carrera; y a todas aquellas personas que de una u otra forma supieron ofrecerme su apoyo incondicional en los momentos más difíciles.

Juan

Dedico este trabajo a todas las personas que de una u otra forma contribuyeron a que logrará esta meta en mi vida.También dedico este trabajo de una forma muy especial a mis padres quienes me guiarón y siempre pensaron que lo lograría

Maritza

El presente trabajo esta dedicado a mi madre la señora Victoria Eleno

Juan

NOMBRE	FIRMA	FECHA
Dr. Romeo Rodríguez DECANO DE LA FACULTAD INFORMATICA Y ELECTRÓNICA	……	……..
Ing. Ivan Ménes DIRECTOR DE ESCUELA DE INGENIERIA EN SISTEMAS	……	……..
Ing.Danilo Pastor DIRECTOR DE TESIS	……	……..
Ing. Wladimir Castro MIEMBRO DEL TRIBUNAL	……	……..
Ing.Landy Ruiz MIEMBRO DEL TRIBUNAL	……	……..
Ing. Eduardo Tenelanda DIRECTOR DEL CENTRO DE DOCUMENTACIÓN	……	……..

NOTA DE LA TESIS ……..

"Nosotros: JUAN BAJAÑA Y MARITZA ORELLANA, somos responsables de las ideas, doctrinas y resultados documentados en esta tesis; y, el patrimonio investigativo e intelectual de la Tesis de Grado pertenece a la **ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**"

_____	_____
JUAN BAJAÑA	MARITZA ORELLANA

INDICE

Indice de tablas

Indice de Figuras

CAPITULO I

1.1 ANTECEDENTES

Todos somos concientes que la Web se está convirtiendo en un elemento clave tanto en el desarrollo de las empresas como de las instituciones, ofreciendo todas ellas información y una amplia gama de servicios a través de la misma.

En estos momentos millones de personas y empresas están ya presentes en la red con sus propias páginas y sus propios nodos Web, mientras que otros cuantos millones se encuentran navegando y haciendo consultas en la red. Durante varios años se ha estado oyendo hablar del valor comercial del Internet, lo que ha provocado que toda empresa que se aprecie disponga de su propio nodo Web. Un gran número de empresas y organizaciones han desarrollado sus propios nodos Web con el fin de tener presencia en la red, para vender productos o servicios o vender espacios publicitarios.

Dado el gran auge que hoy en día tiene Internet, su uso se ha masificado enormemente. Desde páginas meramente informativas hasta sitios interactivos usando tecnologías nuevas.

Actualmente muchos servidores Web colapsan debido a la falta de un análisis previo para su implantación y la gran concurrencia que tienen sus páginas que es a la vez un factor muy importante para la realización de un seguimiento para determinar el rendimiento y funcionalidad que tiene o va ha tener un servidor Web.

Además se necesita saber qué información es recogida por los servidores Web cuando existe una petición a una página determinada, que es lo más importante que se debe medir y como obtener el máximo provecho de los archivos de registro del servidor.

Por esta razón se requiere de un análisis para determinar qué pasos se deben seguir para la optimización de servidores Web en base a los requerimientos o necesidades.

Uno de los proyectos de la Escuela Superior Politécnica del Chimborazo, es el de convertirse en ISP, para de esta manera brindar servicios dial-up a sus usuarios ya que cuenta con la infraestructura para hacerlo.

En la actualidad la ESPOCH cuenta con algunas aplicaciones que hacen uso de los servidores Web, lo que implica que se deba atender a muchas más transacciones provocando el crecimiento del tráfico en la red y el estar preparados para problemas cómo estos motiva que se realice un estudio del servidor Web para su optimización.

1.2 JUSTIFICACIÓN DEL PROYECTO DE TESIS

Dado que cada vez las comunicaciones a través del Web se ven más saturadas, haciendo que nuestros servidores Web queden congestionados y un tanto "agotados" nace la necesidad de que estos sean sometidos a un análisis exhaustivo, que llevará a que nuestro(s) servidor(es) Web sea(n) optimizado(s), en el sentido de hacer que la carga de información sea más ligera, que sus registros sean liberados de ciertas cosas innecesarias, que sus transacciones sean más rápidas.

Con la optimización de los servidores Web se intenta mejorar los tiempos de respuesta, destinado una parte del ancho de banda para uso exclusivo del servidor Web, es decir, que el tiempo entre solicitudes y respectivas respuestas sea corto, ver cuales son los recursos más solicitados, realizar estadísticas de acuerdo a las páginas más visitadas, distribuir de mejor manera la información, generar un mejor reparto de directorios virtuales, distribuir el correo, hacer que mejore el rendimiento del Servidor Web, hacer que las peticiones ya no se hagan a la misma tarjeta y hacer un seguimiento de los paquetes http.

Al hablar de optimización, se hablando específicamente de la parte software, es decir, es decir que en la parte de software se examinara de una manera minuciosa, cada uno de los registros que componen el servidor Web, viendo detalle a detalle cada una de sus líneas y campos, para así, desactivar opciones que hacen que nuestro servidor Web se demore o a su vez tenga que realizar más transacciones a la hora de contestar una petición y junto con otros valiosos datos, ofrecer toda la información posible acerca de quién está utilizando el nodo Web y cómo lo está utilizando.

También se realizara un estudio minucioso del archivo httpd.conf para saber la configuración del servidor, como Dominios Alojados, información de procesos internos del servidor, tiempo que el servidor espera para recibir y enviar peticiones durante la comunicación, ver si el servidor permite más de una petición por conexión, el número máximo de peticiones permitidas por cada conexión persistente, número de segundos que el servidor esperara a la siguiente petición tras haber dado servicio a una petición, antes de cerrar la conexión, el número de procesos creados al arrancar, el número total de los procesos del servidor, número de peticiones de cada proceso hijo antes de morir, puertos por los que el servidor web aceptara las peticiones entrantes, y características del servidor que están disponibles en un directorio en particular.

Al decir optimización no es solo mejorar rendimiento sino también hacer que el servidor sea seguro es por esto que también en esta guía se topará el tema de la seguridad, aquí también se analizara todos los registros, campos y opciones que tengan que ver con este tema. Así como los protocolos que tengan que ver con una comunicación segura.

Viendo la necesidad de la Escuela Superior Politécnica del Chimborazo nos hemos visto en el deber de realizar esta clase de análisis para sus servidores Web, es por esto que se desea proponer una guía referencial que permita al administrador Web, seguir paso a paso las pautas para que su servidor Web sea optimizado de la mejor manera y así darle a la ESPOCH el aseguramiento en su servidor Web en lo que a hardware y software se refiere.
Con esta investigación se logrará conocer a fondo como está constituido un servidor Web, sus registros, sus archivos, su configuración y cómo poder mejorar su rendimiento, seguridad, funcionamiento, haciendo que este sea aceptable y a su vez que se convierta en punto de partida para futuras investigaciones.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL:

Desarrollar una Guía Referencial para diagnosticar y mejorar el rendimiento, seguridad, accesibilidad, aceptabilidad y funcionamiento a través del análisis y optimización tecnológica de los Servidores Web de la ESPOCH.

1.3.2 OBJETIVOS ESPECIFICOS:

- Estudiar las diferentes técnicas, protocolos, registros, archivos de configuración para el adiestramiento en el funcionamiento de los Servidores Web.
- Realizar un estudio minucioso del funcionamiento de los Servidores Web más utilizados (Apache, Internet Information Server).
- Realizar un estudio detallado del rendimiento de los Servidores Web de la ESPOCH.
- Determinar parámetros que permitan mejorar y optimizar un Servidor Web.
- Desarrollar una guía referencial para optimizar servidores Web en base a parámetros de: rendimiento, seguridad, accesibilidad, aceptabilidad y funcionamiento.
- Aplicar la guía realizada a los Servidores Web de la ESPOCH para mejorar su rendimiento, seguridad, accesibilidad, aceptabilidad y funcionamiento.
- Medir y comparar los resultados obtenidos al aplicar la guía realizada a los Servidores Web de la ESPOCH.

1.4 HIPÓTESIS

Al desarrollar una guía referencial para el análisis y optimización de Servidores Web se podrá diagnosticar y mejorar aspectos relacionados con el rendimiento, seguridad, aceptabilidad y funcionamiento.

CAPITULO II

2.1. INTRODUCCIÓN A LOS SERVIDORES WEB

Todos somos concientes que la Web se está convirtiendo en un elemento clave tanto en el desarrollo de las empresas como de las instituciones, ofreciendo todas ellas información y una amplia gama de servicios a través de la misma.

En estos momentos millones de personas y empresas están ya presentes en la red con sus propias páginas y sus propios nodos Web, mientras que otros cuantos millones se encuentran navegando y haciendo consultas en la red. Un gran número de empresas y organizaciones han desarrollado sus propios nodos Web con el fin de tener presencia en la red, para vender productos o servicios o vender espacios publicitarios.

Dado el gran auge que hoy en día tiene Internet, su uso se ha masificado enormemente. Desde páginas meramente informativas hasta sitios interactivos usando tecnologías nuevas.

Siempre se habla de servidores web, pero mucho no tiene claro lo que abarca la conceptualización de servidor web y que archivos la conforman. Para ello vamos hacer una revisión breve de los servidores web: sus registros, archivos de configuración y la clase de seguridad que nos brindan para proteger nuestra información.

2.2 SERVIDORES WEB, REGISTROS, ARCHIVOS DE CONFIGURACIÓN.

2.2.1 SERVIDOR WEB.

Definición de Servidor Web

Existen muchas definiciones de lo que es un servidor Web. Al efectuar la investigación se trato de hacer un compendio de todas estas definiciones. A continuación se mostrará algunas de las definiciones más usadas.

Un servidor Web es un programa que implementa el protocolo HTTP (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML (hypertext markup language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos.

Sin embargo, el hecho de que HTTP y HTML estén íntimamente ligados no debe dar lugar a confundir ambos términos. HTML es un formato de archivo y HTTP es un protocolo.

Cabe destacar el hecho de que la palabra *servidor* identifica tanto al programa como a la máquina en la que dicho programa se ejecuta. Existe, por tanto, cierta ambigüedad en el término, aunque no será difícil diferenciar a cuál de los dos nos referimos en cada caso. En este artículo nos referiremos siempre a la aplicación.

Un servidor Web se encarga de mantenerse a la espera de *peticiones HTTP* llevada a cabo por un *cliente HTTP* que solemos conocer como *navegador*. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. A modo de ejemplo, al teclear *http://www.espoch.edu.ec/* en el navegador, éste realiza una petición HTTP al servidor de dicha dirección. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla. Como vemos con este ejemplo, el cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma.

Sobre el servicio Web *clásico* podemos disponer de aplicaciones Web. Éstas son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- *Aplicaciones en el lado del cliente*: el cliente Web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o Javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas *scripts*). Normalmente, los navegadores permiten ejecutar aplicaciones escritas en

lenguaje *javascript* y *java*, aunque pueden añadirse mas lenguajes mediante el uso de *plugin*s.

- *Aplicaciones en el lado del servidor*: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Las aplicaciones de servidor suelen ser la opción por la que se opta en la mayoría de las ocasiones para realizar aplicaciones Web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad adicional, como sí ocurre en el caso de querer ejecutar aplicaciones javascript o java. Así pues, cualquier cliente dotado de un navegador Web básico puede utilizar este tipo de aplicaciones.

El servidor Web es un programa que corre sobre el servidor (físicamente hablando) que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición.

El servidor Web va a ser fundamental en el desarrollo de las aplicaciones del lado del servidor, *server side applications*, que vayamos a construir, ya que se ejecutarán en él.

Para conocer más de los registros del servidor hablaremos brevemente cuales son y para que sirven.

Registro del Servidor Web.

Los registros de los servidores web son archivos internos donde se guarda toda la información de la actividad que este ha tenido, esa información es muy importante al momento de hacer una revisión del servidor y tomar decisiones acerca del mismo.

Los desarrolladores de los servidores web hacen todo lo posible por esconder la información en bruto que se encuentra almacenada en estos archivos de registro, sin embargo, es fundamental tener conocimiento mínimo de estos archivos, para ver que es posible aprender y deducir de los datos almacenados en los servidores web.
A continuación mencionaremos cuales son y para que sirven estos archivos de registro.

El Registro de Transferencia.

A veces llamado registro de acceso, se podría decir que este es el registro más importante del servidor, ya que en él se puede observar detalladamente que es lo que esta ocurriendo en el servidor. Cada acceso que recibe el servidor se anota en el registro de transferencia junto con la fecha y hora de la petición, el nombre o dirección IP de la computadora que realizó la petición, el texto de la petición, un código de estado y el número de bytes enviados al peticionario, etc.

El Registro de Error.

Cuando se produce un error o devuelve un código de error en una transacción http, el servidor, generalmente, registra información sobre la transacción tanto en el registro de transferencia como en el registro de error, digo generalmente porque no todos los resultados se almacenan en ambos registros.

Entre los errores que se almacenan en ambos registros se encuentran las peticiones de páginas que no existen, intentos de acceder a archivos y páginas para los cuales el lector no tiene permiso y una gran variedad de posibles condiciones de error.

Al igual que el registro de transferencia el registro de error almacena la fecha en cada una de las líneas del archivo de registro.

El Registro de Remite.

Este es uno de los dos archivos de registro opcional. El servidor web utiliza el registro de remite para anotar el URL desde el que se realizan las peticiones. Dicho con otras palabras, registra de donde vienen los lectores o como han encontrado las páginas. Esta es una información tremendamente útil cuando se desea atraer el tráfico hacia un nodo.

El registro de remite incluye el URL desde el que proviene el lector y la página web a la que a llegado después de seguir el enlace.

El Registro de Agente.

Se utiliza para anotar el nombre y la versión del agente de usuario que efectúa las peticiones. El agente por lo regular es un visualizador de web. Cuando un visitante malintencionado satura con peticiones un nodo web, deja su marca en los archivos de registro igual que lo haría un lector cualquiera.

Cabe recalcar que los archivos de registro son muy importantes en el estudio de los Servidores Web, pero no sólo los archivos de registros existen en los servidores, sino también los archivos de configuración, ya que a través de ellos podemos adaptar nuestro servidor a la forma como aparezcan nuestras necesidades.

Conozcamos algunos de estos archivos que nos serán útiles al momento de configurar nuestro servidor web.

Archivos De Configuración Del Servidor

A continuación se da una descripción general de las funciones generales de cada uno de estos archivos.

httpd.conf

En este archivo se indica la configuración principal del servidor. Aquí se configuran atributos como la asignación del puerto para el servidor, el dueño bajo el que se ejecuta el servidor, etc. También se indican aquí, parámetros que controlan la ejecución de servidores httpd paralelos.

srm.conf

Aquí se ajustan parámetros como la raíz del árbol de documentos, funciones especiales como SSI, manejo de los mapas sensitivos, etc.

access.conf

Gestiona restricciones de acceso al servidor.

mime.conf

Específica asociaciones entre tipos MIME conocidos y extensiones de archivo.

Directivas contenedoras (en access.conf)

Se refieren a un determinado directorio o conjunto de archivos, y se usan para englobar o incluir otras.

Directivas para restricciones de acceso (en access.conf)

Las más comunes (dentro de una directiva contenedora) son:

- *Options:* permite indicar opciones disponibles en un directorio, como la posibilidad de ejecutar CGI's, incorporar SSI, etc.

- *order:* indica el orden en que se evaluarán las directivas *allow* y *deny*
- *allow from, deny from*: especifican una máscara de máquinas a las que se permitirá o denegará el acceso al directorio.

Directivas para CGI (en srm.conf)

Permite designar un directorio para que ejecute programas CGI, con la directiva ScriptAlias.

Asociación de tipos MIME (archivo mime.types)

Los tipos **MIME** conocidos y sus extensiones asociadas se encuentran en el archivo mime.types.

Restricciones de acceso (archivo access.conf)

Es posible restringir el acceso a determinados directorios en función del origen de la petición y también se pueden emplear identificaciones mediante *login* y *passwords.*

2.3 SEGURIDAD, PROTOCOLOS DE COMUNICACIÓN SEGUROS.

Seguridad

Cuando un sistema es usado como un servidor en una red pública, se convierte en un objetivo para ataques. Por esta razón, es de suma importancia para el administrador fortalecer el sistema y bloquear servicios.

Antes de extendernos en problemas particulares, debería revisar los siguientes consejos generales para mejorar la seguridad del servidor:

- Mantenga todos los servicios actualizados para así protegerse de las últimas amenazas informáticas.
- Utilice protocolos seguros siempre que sea posible
- Proporcione sólo un tipo de servicio de red por máquina siempre que sea posible.
- Supervise todos los servidores cuidadosamente por actividad sospechosa.

Protocolos De Comunicación Seguros

SSH

SSH (o Secure SHell) es un protocolo para crear conexiones seguras entre dos sistemas. Usando SSH, la máquina del cliente inicia una conexión con una máquina de servidor. SSH proporciona los siguientes tipos de protección:

- Después de la conexión inicial, el cliente puede verificar que se está conectando al mismo servidor durante sesiones ulteriores.
- El cliente puede transmitir su información de autentificación al servidor, como el nombre de usuario y la contraseña, en formato cifrado.
- Todos los datos enviados y recibidos durante la conexión se transfieren por medio de encriptación fuerte, lo cual los hacen extremamente difícil de descifrar y leer.
- El cliente tiene la posibilidad de usar X11 aplicaciones lanzadas desde el indicador de comandos de la shell. Esta técnica proporciona una interfaz gráfica segura (llamada reenvío por X11).

El servidor también obtiene beneficios por parte de SSH, especialmente si desempeña una cierta cantidad de servicios. Si usa el reenvío por puerto, los protocolos que en otros casos serían considerados inseguros (POP, por ejemplo) se pueden cifrar para garantizar comunicación segura con máquinas remotas. SSH hace relativamente sencilla la tarea de cifrar tipos diferentes de comunicación que normalmente se envía en modo inseguro a través de redes públicas.

SSL

El protocolo ssl es un protocolo que se sitúa entre el protocolo de la capa de red (ej: TCP/IP) y un protocolo de la capa de aplicación (ej: HTTP). Ssl proporciona mecanismos para establecer una comunicación segura entre un cliente y un servidor, por medio de autenticación, el uso de firmas digitales para validar integridad y el uso de encriptación para privacidad.

Este protocolo esta diseñado para soportar un rango de algoritmos de criptografía, algoritmos de digest y signatures. Esto le permite a los servidores elegir que tipo de algoritmo va a utilizar y además toma ventaja de futuros nuevos algoritmos. Las opciones se negocian entre el cliente y el servidor al inicio de la sesión.

HTTPS

Uno de los usos comunes de ssl es el de establecer una comunicación Web segura entre un browser y un Webserver. Es aquí donde se usa https que es básicamente http sobre ssl con un esquema de invocación por medio de url. Es importante hacer notar que el uso del protocolo https no impide en caso alguno que se pueda utilizar http, por lo que la mayoría de los browsers advierten cuando una página tiene

elementos que no son seguros en entornos seguros, como también advierten cuando se invoca un protocolo distinto al de la pagina actual (http -> https o https -> http).

2.4 SERVIDORES WEB MÁS UTILIZADOS.

Servidores Web Más Utilizados

Los servidores HTTP más conocidos (en el "dominio público"), son los siguientes:

APACHE

Desde 1996, es el servidor más expandido. Actualmente, más de 52% de los servidores WWW en el mundo utilizan el programa Apache (porcentaje en continuo aumento). Al nivel de la configuración, es un servidor enteramente compatible con el servidor NCSA, pero además rápido. Por lo tanto es muy fácil migrar del servidor NCSA hacia el servidor Apache. Para obtener mejores performances, el servidor Apache guarda continuamente una serie de procesos creados previamente de manera a estar listos para responder a un requerimiento desde que éste llega, mientras que el servidor NCSA crea un proceso en el momento en el cual un requerimiento es recibido.

Permite utilizar los mismos scripts CGI que con el servidor NCSA. Desde hace poco (1998), una versión para Windows 95 y NT está disponible. Un interfaz para OpenSSL permite añadir un mecanismo de ocultación (SSL y TLS) garantizando la confidencialidad y la seguridad de la comunicación entre servidor y navegador. Está disponible en Linux.

IIS (Internet Information Server)

Es un servidor web, que incluye los servicios de: HTTP, HTTPS, FTP, SMTP (correo saliente) y NNTP (grupos de noticias). Además es capaz de ejecutar varios motores de script como: ASP, PHP, Cold Fusion, etc.

Esta interfase permite programar aplicaciones del lado del servidor en casi cualquier lenguaje, pero los más utilizados son VBScript (una versión reducida de Visual Basic) y JScript (una versión de JavaScript). Lo interesante de ASP es que también funciona sobre Windows 95/98 con el **Microsoft Personal Web Server**, incluido gratis en uno de los Service Pack de Microsoft en su web. Esto permite realizar pequeñas aplicaciones en redes con o sin Windows NT, intranets o en la computadora local.

CERN

Es el primer servidor en introducir la noción de **proxy** que permite a este servidor relevar los requerimientos de "clientes" que se hallan detrás de las barreras de acceso ("firewall") y a manejar un cache de los documentos encontrados.

PLEXUS

Escrito en Perl. Ofreciendo mucha flexibilidad de programación, sobretodo ha sido utilizado hace varios años atrás. Pero siendo menos rápido que los otros programas servidor, cada vez es menos utilizado.

WN

Este servidor para estaciones Unix incorpora la posibilidad de efectuar búsquedas de documentos, así como la posibilidad de combinar todo un conjunto de documentos como un solo documento virtual a fin de facilitar la impresión. La búsqueda de documentos puede hacerse sobre la base de los títulos, de palabras-claves, o incluso del contenido completo de los documentos.

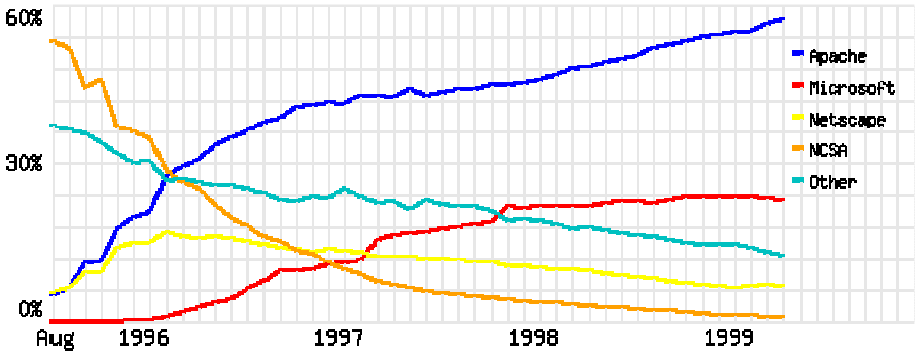

Figura 1 Grafico de Estadisticas de servidores web más utilizados

Nombre. Servidor	Número de Usuario	Evolución/Mes
Apache	3'097'993	57.22%
IIS	1'169'415	22.99%
Netscape	341'876	6.78%

Tabla 1 Servidores Web más utilizados

Aún hay bastantes servidores, como PHTTPD, Pictorius NetServers (para Mac). Netscape y Microsoft tienen por supuesto, cada uno su propio programa servidor, pero estos programas son pagables. La utilización de un programa pagable no se justifica a menos que se quiera hacer comercio electrónico vía Web. En junio de 1997, los servidores más utilizados a través del mundo eran:

Como se sabe, dos de los servidores Web más utilizados son el Apache y el Internet Information Server.

2.5. ANÁLISIS DE LOS REGISTROS Y LOS DATOS DEL SERVIDOR

2.5.1. REGISTROS DE ACCESO DEL SERVIDOR

Cuando un cliente http se conecta a un servidor web (sea este apache, IIS, etc.), se intercambia mucha información, toda esta información, es registrada o almacenada en los registros del servidor. Cuando se instala cualquier servidor web, se crea por defecto el registro de transferencia, el mismo, que veremos en detalle mas adelante. Al decir que solo se crea el registro de transferencia no quiere decir que sea el único registro, sino por el contrario, que se puede aumentar el número de los registro a medida que aparezcan las necesidades.

Este aumento de registros es posible en los servidores IIS y Apache, que son nuestro caso de estudio.

Existen cuatro registros principales que van ha ser objeto de estudio en esta etapa del proyecto, de los cuales vimos al inicio de forma general cada uno de ellos y para qué sirven, ahora hablaremos de una forma detallada y los parámetros que lo conforman.

Con esto lograremos que se conozca a fondo cada uno de los registros del servidor para posteriormente realizar el análisis de la información contenida en los mismos.

Hay que recalcar que los nombre utilizados a continuación para cada registro, son nombres específicos de acuerdo a su información y a la función que desempeñan.
La siguiente descripción de los registros es de forma general sin mencionar detalles de un servidor web específico.

2.5.1.1 EL REGISTRO DE TRANSFERENCIA

Como habíamos visto anteriormente que en el registro de transferencia se anotan todos los accesos al servidor web, incluyendo todas las consultas realizadas correctamente al servidor web especificando parámetros como: la fecha y hora de la petición, el nombre o dirección IP de la computadora de la que ha partido la petición, el propio texto de la petición, un código de estado, el número de bytes enviados al peticionario.

Por cada acceso el servidor añade una línea de transferencia para registrar de donde proviene la petición, el formato más común del registro de transferencia se denomina, por razones obvias, *Formato Común de Registro.*

Estructura del registro de transferencia:

Registro de Transferencia		
El Campo Host		
El Campo RFC 931		
ID Usuario	**Fecha**	
El Campo petición http		
Estado	**Volumen Tr.**	**Opcional**

Tabla 2 Registro de Transferencia

El registro de transferencia, contiene información básica de cada transacción http que maneja el servidor. Esto se puede emplear para generar informes estadísticos acerca del tipo de patrones de uso que ve su sitio web. También es posible generar registros de transferencias personalizados para recoger información específica que le pueda interesar.

Parámetros del Registro de Transferencia

El Campo Host

Este es el primer campo del registro de transferencia. En la mayoría de las ocasiones es el nombre cualificado del host remoto que establece una conexión y solicita un documento; en otras, será únicamente la dirección IP del cliente que establece la conexión.

Si usted es el administrador del servidor web de su empresa, y necesita conocer quien está o ha visitado su nodo, en este campo host del registro de transferencia la aparecerá información que le será de mucha ayuda para esa tarea, como por Ejemplo, parte de la información útil que podrá ver en este campo es, el nombre del dominio cualificado del computador o la dirección IP del visitante, como por ejemplo, mostraremos dos entradas a este campo.

Pc36.decitel.espoch.edu - - [28/Oct/2005:12:55:25 -0700] “GET / HTTP1.0” 200 5138

165.257.49.151 - - [28/Oct/2005:12:55:25 -0700] “GET / HTTP1.0” 200 5138

Ciertamente, ninguna de las dos entradas nos revela demasiada información. Sin embargo, con la primera de ella se puede observar que la petición procede de alguien que se encuentra delante de la una PC llamada "Pc36" en el departamento llamado Desitel de la Escuela Superior Politécnica del Chimborazo. Con solo esto ya se tiene mucha más información que la proporcionada únicamente por la IP, al menos a simple vista.

Recomendación

En algunas ocasiones, puede ser necesario ***desactivar el registro de nombre de dominio*** especialmente si el servidor soporta bastante tráfico. La razón es que el servidor tiene que hacer mucho más trabajo para anotar este nombre, ya que debe utilizar el Sistemas de Nombrado de Dominio (DNS) para encontrar el nombre del host remoto. Los paquetes de datos que transportan la petición desde la computadora remota, sólo contienen realmente la dirección IP. Para convertir la dirección IP en un nombre de computadora, el sistema tiene que hacer una petición al DNS de la Red.

Normalmente, esta petición solo consume unos pocos milisegundos en la mayoría de las redes, pero en un servidor muy cargado, estos pequeños espacios de tiempo de CPU y el ancho de banda, puede incrementar considerablemente los tiempos de acceso. De todas formas, si no se realiza una búsqueda DNS tampoco se pierde nada de información, ya que se mantiene la dirección IP del sistema remoto. Por ejemplo, si se desea realizar un análisis estadístico de las personas que más nos visitan lo podemos hacer por la dirección IP, ya que las herramientas utilizadas para analizar las estadísticas del servidor seguramente están preparadas para trabajar con direcciones IP de la misma manera que con nombres de computadoras.

Para terminar haremos la última acotación con referencia a este campo, si se obtiene en nombre del computador remoto, puede ser que falle al obtener la dirección IP, esto es así porque no se asigna un nombre permanente a cada una de las dirección IP existentes en el Internet. Una razón para ello es que muchos ISP's asignan a sus clientes las direcciones IP de forma dinámica. Esto quiere decir que cada vez que uno de estos clientes enciende su computador para navegar en la red, va a obtener una dirección IP diferente a la que obtuvo la última vez que se conectó.

El resultado de todo esto es que no se va a obtener un nombre de computadora en el registro de transferencia, si no una dirección IP. Al analizar los archivos de registro, lo

mejor que se puede hacer es obtener el nombre y la localización geográfica del ISP al que pertenece dicha dirección IP.

El Campo RFC 931.

El segundo campo del registro de transferencia se denomina Campo RFC 931 o Campo de Identidad. Lo mas importante que se debe de tener en cuenta con este campo es que casi siempre es un guión (-). Esto es así por que casi nunca se lo utiliza.

El propósito de este campo es permitir al administrador Web conocer la identidad real de la persona (al menos su nombre de usuario) que solicita una de las páginas Web. El problema es que para obtener dicha información, la computadora del otro extremo debe de estar ejecutando un programa (otro demonio del servidor) que atienda las peticiones del nombre del usuario y la responda.

Conclusión

El protocolo de identificación no se ha desarrollado como un protocolo de autorización o control de acceso. En el mejor de los casos, proporciona información administrativa adicional relativa a conexiones TCP. En el peor de los casos, proporciona información errónea, incorrecta o intencionadamente incorrecta.

El Campo Identificación de Usuario

La mayoría de los Servidores Web implantan un subsistema de identificación de usuario denominado simplemente ***Identificación Básica de Usuario.*** Dentro de este sistema, el administrador de un nodo web puede decidir si permite el acceso a todos los usuarios o únicamente a ellos usuarios autorizados y a cuales de los directorios del servidor web en particular. Para que el administrador pueda activar la identificación de usuario debe situar un archivo especial en unos subdirectorios protegidos, crear un archivo de Base de datos de contraseñas que contenga los nombres de los usuarios y sus contraseñas cifradas, y ponerlo todo junto en los archivos de configuración de ejecución del Servidor.

Una vez que se protege un directorio de esta manera, si un lector, mientras consulta una página web, trata de acceder a una archivo de ese directorio, tendrá que realizar correctamente los pasos del proceso de identificación antes de que el servidor le permita seguir adelante. Prácticamente todos los Servidores web actuales saben como

realizar este procedimiento. El visualizador despliega un cuadro de dialogo con un mensaje que solicita el nombre de usuario y contraseña del lector.

El lector deberá introducir su nombre de usuario y contraseña y pulsar el botón aceptar, para que el visualizador envíe los datos del usuario y contraseña al Servidor.

El servidor compara el nombre que ha introducido el lector con su lista de contraseñas y cifra la contraseña enviada con el mismo algoritmo utilizado inicialmente para cifrar la contraseña.

Si el nombre del usuario es valido y la contraseña recién cifrada coincide, el servidor transferirá la página solicitada junto con una clave de identificación que almacenara el visualizador y que enviara al servidor cuando el lector desee acceder a otros documentos protegidos. De esta forma, el lector no tiene que realizar el proceso de identificación por cada uno de los documentos de la jerarquia protegida.

Conclusión

El tercer campo del registro de transferencia se llama Campo de Identificación de Usuario. Este campo toma el valor del nombre del usuario (sin la contraseña) que introduce el lector. Por ello, de alguna manera, este campo es similar al campo RFC 931, excepto en que el nombre de usuario no es el nombre del lector en su computadora, sino el nombre del usuario de una determinada porción de un nodo web, como una base de datos de información confidencial. En este campo, es lector el que introduce el nombre de usuario directamente en vez de ser el servidor el que determina el nombre solicitándolo por la puerta trasera.

Aunque la identificación básica de usuario mantendrá a raya a la practica de zurríos no autorizados, no resulta un trabajo excesivamente complicado, para un pirata informático experto y decidido, romper la protección.

Cuando se escribe el nombre de usuario y la contraseña en el cuadro de diálogo de identificación de usuario, el visualizador no cifra las cadenas de caractéres de forma segura antes de enviarla al servidor web.

El Campo Fecha.

El campo menos significativo del registro de transferencia con Formato común de registro, pero no por ello menos útil, el campo fecha, y consta de tres partes:

- Fecha
- Hora
- Deferencia con la hora de Greenwich (GMT).

Entre las tres, se obtiene la hora local del servidor que solicita la petición, lo que nos permite realizar una importante observación.

Puesto que los servidores web no suelen moverse, ¿Por qué es importante indicar la diferencia con respecto a la hora de greenwich en cada acceso? ¿No debería ser siempre la misma (excepto en una diferencia de una hora entre el horario de verano y el de invierno)? ¿No es redundante? ¿Cuántos terabytes en disco se están mal gastando en todo el mundo por tener que almacenar esto seis bytes de datos adicionales por todos y cada uno de los accesos? ¿Quién lo sebe? (Quizá la gente que aprobó el Formato común de registro tenía acciones de las empresas dedicada a la elaboración de discos fijos.).

A continuación se presenta una utilización aceptable de la diferencia GMT. Un nodo web particularmente grande puede consistir en una colección de nodos replicados geográficamente muy distantes. Por ejemplo, un nodo web de la ESPOCH puede tener espejos en Panamá y Canadá. Suponiendo que el administrador de la ESPOCH, quisiera realizar estadísticas de todo el nodo, la diferencia GMT tendría mucho valor para mantener la zona horaria bien definida.

El Formato común de registro especifica que la fecha debe encontrarse en el siguiente formato:

DD/Mes/AAAA

DD es el día del mes en número (precedido por cero si el número va de uno a nueve), Mes son las tres primeras letras del nombre del mes (por ejemplo: Jun, Oct) y AAAA son los cuatro caracteres numéricos del año.

Inmediatamente después de la fecha se encuentra dos puntos y la hora con este formato:

HH:MM:SS

HH es la hora, MM son los minutos y SS es el número de segundos.
Finalmente, la última parte del campo fecha es la diferencia GMT. Después de la hora, hay un espacio y un signo más o menos que indica el número de horas de diferencia GMT (multiplicado por cien por alguna oscura razón). Por ejemplo en horario de verano en el Pacífico (PDT) es siete horas al oeste (o siete horas menos) que la hora GMT. Por ello, la diferencia GMT se indica en el archivo de registro como -0700.
A continuación se muestra el aspecto real del campo fecha:

[10/Jun/2005:12:58:25 -0700]
[10/Jun/2005:13:58:25 -0700]

El Campo Petición http.

Aquí se encuentra la sustancia de la transacción, la petición real del usuario remoto. Este es el quinto campo del registro de transferencia con Formato común de registro.

Es el único campo que está delimitado por comillas, puesto que es el único campo que puede tener espacios (que en el resto de casos sirve para campos).

La petición http comienza por un método clave (u orden o petición, dependiendo del punto de vista de cada uno). Hay tres posibles métodos:

- GET
- POST
- HEAD

El método GET indica al servidor que el lector está solicitando un documento o está especificando el nombre de un programa que realiza algún tipo de procesamiento y producirá una salida que deberá <<devolver>> al cliente. Este es el método estándar para solicitar una página.

La mayoría de los visualizadores mantiene una caché en disco con las páginas y gráficos consultados más recientemente. Si el visualizador determina que la página solicitada se encuentra en su cache de disco, añadirá una cabecera especial a la petición http, llamada If-Modified-Since. El dato de esta cabecera es la fecha del archivo en la cache de disco. Esto realmente convierte a la petición GET estándar en una petición GET condicional. El servidor comprueba si la página o gráfico se ha

modificado después de esa fecha y únicamente se enviará si ha sufrido cambios. En caso contrario devuelve al visualizador un código de estado especial.

El método POST indica al servidor que se van ha enviar unos datos a continuación y se supone que el URL suministrado es un programa o script que puede aceptar y utilizar dichos datos. Cuando se rellenan los campos de información de un formulario en una página web y se pulsa en el botón Submit (Enviar), se está solicitando normalmente una acción POST.
El método HEAD es similar al método GET, excepto en que el servidor únicamente devuelve la sección <<HEAD>> de cualquier documento (junto con la cabecera http estándar inherente a toda transacción http). Este método es útil para comprobar la validez de los enlaces de hipertexto, normalmente de forma automática. (Realmente no se puede hacer que un visualizador web estándar genere peticiones HEAD. Con un visualizador, o se desea ver una página o no se desea. Las peticiones HEAD se generan habitualmente con otro tipo de programas.).

La última parte de la petición http es el nombre y número de versión del protocolo. Casi siempre suele ser http/1.0, aunque en algunas ocasiones se puede ver http/0.9. Por descontento, ya existe una nueva versión del protocolo http (versión 1.1).

El Campo Código de Estado.

El penúltimo campo del registro de transferencia es el campo código de estado. Este es un código que genera el servidor (tanto como para escribir en sus archivos de registro como para responder al cliente remoto). El código de estado describe el éxito o fracaso de la transacción.

Todos los códigos de estados consisten en un número de tres dígitos, que se transmite como tres caracteres ASCII. Hay cuatro clases de código de estado que es importante conocer.

Clase	**Significado**
200	Éxito
300	Redirección
400	Fallo
500	Errores del servidor

Tabla 3 Clases de código de estados

Los código de estados que se encuentran dentro de cada una de las categorías (o clases) existentes tienen números pertenecientes al mismo rango. Esto es, todo los códigos de éxito empiezan por 200, los código de redirección empiezan por 300, etc. También hay una clase de estado 100 no utilizada todavía. Afortunadamente. No hay demasiados códigos de estado en http versión 1.0 (solo 15 en total). Sin embargo la versión 1.1 del protocolo define mucho más.

Código	**Descripción**
200	Ok
201	Creado
202	Aceptado
204	Sin contenido

Tabla 4 Código de Estado de éxito

Esta tabla muestra los códigos de éxito. El código 200 representa básicamente la obtención correcta de una página web, aunque este código también se devuelve al efectuar interacciones correctas con otras entidades, como programas CGI. Este es el código de estado que se puede ver con más frecuencia en el registro de transferencia.

El código de estado 201 y 202 son relativamente raros. De hecho, es posible que no los vea nunca, por que actualmente no hay aplicaciones para ellos. Sin embargo, si podrá ver en algunas ocasiones el código 204. Esto indica normalmente que un visualizador o programa CGI ha generado una cabecera http incorrecta.

Código	**Descripción**
301	Se ha movido permanentemente
302	Se ha movido temporalmente
304	No se ha modificado

Tabla 5 Código de Estado de redirección

El código 301 es poco frecuente, un administrador web puede crear una entrada en los archivos de configuración de ejecución del servidor (con la directiva Redirect) para que el servidor responda a las peticiones de un determinado URL con código 301. Esto se hace para notificar al lector que un recurso se ha movido permanentemente a otro lugar, aunque no todos los visualizadores lo soportan. Los que no lo soportan muestran el familiar mensaje <<Este documento ha cambiado de lugar>>.

Código	Descripción
400	Petición errónea
401	No autorizado
403	Prohibido
404	No encontrado

Tabla 6 Código de Estado de Fallo

El código de fallo probablemente más conocido es el código de estado 404 (no encontrado). Cada vez que se introduce un URL equivocado o se sigue un enla ce anticuado, aparece este código tan habitual. El código de error 400 indica que el lector ha escrito un URL tan incorrecto que el servidor ni siquiera a sido capaz de determinar que es lo que se solicitaba.

El código 401 indica que es necesaria una identificación de usuario para acceder al objeto solicitado y el usuario no ha conseguido identificarse correctamente con el par nombre de usuario/contraseña.

El código 403 significa que el servidor comprende la petición pero no puede satisfacerla. Por ejemplo, si el usuario trata de obtener un índice del directorio cgi-bin, o de otro de los directorios para los que el servidor no dispone de los permisos adecuados, algunos servidores no realizarán la petición y devolverán un código de estado 403.

Desgraciadamente, no se puede esperar que todos los servidores web reaccionen uniformemente cuando se produce una condición de error. Por ejemplo, cuando el servidor Apache, trata de acceder un directorio para el que no se dispone de los permisos adecuados, genera un redirección (estado 302) a ningún sitio en vez de responder con un código de error.

Código	Descripción
500	Error interno del servidor
501	No implementado
502	Pasarela incorrecta
503	Servicio no disponible

Tabla 7 Código de estado de error del servidor

Los códigos de estados de error del servidor. En una palabra, si aparece alguno de estos códigos en el registro de transferencia, tendrá que ponerse en el papel de administrador y buscar el problema, porque algo está mal.

Lo que se debe recordar de los errores (tanto fallos como errores del servidor) es que no van a formar parte de las estadísticas del nodo web. Son una indicación administrativa de alarma para indicar que algo está mal en la configuración del servidor o en los programas CGI, o que alguien está tratando de acceder a objetos a los no debería acceder.

Recomendación

Si está realmente interesado en hacer un seguimiento completo a todos los errores, la mejor política es conocer el servidor web en particular y saber como reacciona a cada uno de los posibles tipos de error. Así, cuando analice los archivos de registro (o los prepare para un análisis automatizado), siempre comprenderá mejor la información obtenida.

El Campo Volumen Transferido.

El último campo del registro de transferencia es el volumen transferido. Es una representación ASCII del número de bytes transferidos entre el servidor y el cliente como resultado de una petición http.

Puesto que las únicas transacciones que transfieren son las peticiones GET que devuelven un código de estado 200 (Ok), el campo volumen de transferencia no es utilizable en el resto de de anotaciones del registro, y contendrá un guión (-) o un cero (0). Por ejemplo, si un agente de usuario realiza una petición GET condicional de un documento que no se ha modificado, el servidor anotará un estado 304 (No modificado) y el campo volumen transferido será únicamente un guión (-). Por el contrario, una petición HEAD correcta, producirá un estado 200 (Ok) pero mostrará cero (0) bytes transferidos.

Campos Opcionales del Registro de Transferencia.

Existen dos archivos de registros adicionales que complementan la información del registro de transferencia. Estos son: el registro de remite y el registro agente. El registro de remite registra el URL de la página web o el recurso que ha seguido un usuario para llegar al nodo web. Esto puede ser una pagina de otro nodo web con un

enlace hipertexto a una pagina de este nodo web, o el URL de un motor de búsqueda junto con los criterios de la búsqueda que ha introducido el lector para encontrar un listado de nodos.

El registro de agente anota el nombre del visualizador web que utiliza el lector y su número de versión (y a veces, el sistema operativo).
Una extensión popular del Formato común de registro es el Formato de registro combinado o extendido. El Formato de registro combinado integra los registros de remite y agente dentro del registro de transferencia normalmente, cuando se hace esto, el registro de transferencia ya no tiene un formato común de registro, aunque el formato de registro combinado puede resultar mas eficaz y su utilización mas deseada. Por fin, con el Formato de registro combinado, se puede hacer coincidir la información de remite y de agente de usuarios con los accesos a los que se refiere.

2.5.1.2 EL REGISTRO DE ERROR

El registro de error añade una línea nueva por cada fallo o condición de error que se detecta en el servidor. Esencialmente, el registro de error tiene dos campos: una fecha y una descripción del error o fallo.

El Campo Fecha

El primer campo del registro de error es la fecha. Es muy parecida a la fecha de registro de transferencia, pero, inexplicablemente, tiene un formato diferente. El formato de la fecha del registro de error es el siguiente:

[Día Mes D HH:MM:SS AAAA]

Día Son los tres primeros caracteres del día de la semana [por ejemplo Sat (Sábado), Wed (Miércoles)] y D es el día del mes (no necesariamente precedido por cero si es menor que diez).

Conclusión

La explicación anterior indica que si se hace una ordenación ASCII del contenido del registro de error las entradas que se hayan producido en viernes irán delante de las que se hayan producido el lunes o de cualquier otro día de la semana, independientemente de la semana o del mes.

Esto trae a colación otra pregunta: ¿Cómo puede ser que los diferentes servidores utilicen formatos distintos? .El formato común de registro, solo define los contenidos y formatos del registro de transferencia. No hay ninguna regla o indicación para el formato del resto de los archivos de registro, y en caso de que lo hubiera, no hay nadie encargado de hacerlo cumplir.

Mensajes del Registro de Error

Por cada uno de los códigos de error o fallo, se genera una línea en el registro de error. En el registro de error, los errores indican, no con su código (como en el registro de transferencia) sino con una descripción textual del error. Por ello, una forma de entender el registro de error, es verlo como si fuera una versión legible de los errores notificados en el registro de transferencia. Desgraciadamente, los diferentes formatos de fecha hacen muy difícil la comparación de las entradas. Los mensajes de error del registro de error de pueden clasificar en varias categorías: mensajes administrativos, fallos de acceso, conexiones pérdidas y fallos por exceso de tiempo.

Mensajes administrativos

Los mensajes administrativos son textos informativos anotados en el servidor, completamente independientes de los accesos al nodo que hacen los lectores. Entre ellos se encuentran:

- Mensajes de error del servidor.
- Señales que recibe el servidor.
- Mensajes de terminación del servidor.

Cuando se arranca el servidor por primera vez una de las primeras cosas que hace es crear los archivos de registro. Inicialmente, estarán todos vacíos. Antes de que el servidor se ponga a la espera de recibir conexiones http, utilizara el registro de error para indicar que ha arrancado correctamente (o incorrectamente). La línea de arranque del servidor tiene el siguiente aspecto:

[Sat Jul 6 13:06:45 1996] Server configured - resuming normal operations

Fallos de Acceso

El resto de mensajes de error se puede clasificar en fallos de acceso y conexiones perdidas. Los errores de acceso se corresponden directamente con la clase de código de estado 400. Puesto que el estado 404 (No encontrado) es el que va a aparecer con mas frecuencia.

Debemos destacar que, cuando un lector solicita una página que no existe, el servidor devuelve un código 404 a su visualizador web y además registra el intento fallido en el registro de error. La entrada del registro de transferencia tiene el aspecto siguiente:
winpc.crs.com - - [08/Jul/1996:09:55:00 -0700] "GET /events.html HTTP/1.0" 404 –

El código de estado 404 indica que el archivo solicitado (/events.html) no existe y el guión del ultimo campo indica que no se ha transferido ningun byte.

Además de introducir la línea anterior en el registro de transferencia, el servidor escribe la correspondiente línea en el registro de error. Esta entrada tiene un aspecto muy diferente:
[Mon Jul 8-.09:55:00 1996] access to /var/www/docs/events.html failed for winpc.crs.com, reason: File does no exist

[Mon Jul 8-.09:55:00 1996] ha fallado el acceso a /var/www/docs/events.html del nodo winpc.crs.com, motivo: el archivo no existe

Lo primero que se puede observar es que el primer campo del registro es la fecha, al contrario que el registro de transferencia, cuyo primer campo es el nombre del host. El resto del campo es esencialmente una frase larga que describe la naturaleza del error. No es una frase excesivamente concisa, sino más bien explicativa. Como ya se ha indicado, esto es así porque el registro de error se diseño inicialmente para los lectores humanos, en vez de ser un formato de computadora preparado para ser leído con un programa informático.

Conexiones Perdidas

Otro incidente que aparece frecuente en el registro de error son las conexiones perdidas. Estas, generalmente, son los resultados de cancelar la transferencia de una página cuando el lector pulsa el botón stop (detener) o el botón flecha de retroceso en el visualizador, o cuando pulsa sobre uno de los enlaces de la nueva página antes de que esta y todos sus gráficos se hayan transferido completamente, o cuando cierra el visualizador a mitad de una transferencia.

La mayoría de los servidores web no registran las conexiones perdidas en el registro de transferencia. Este error solo aparece en el registro de error.

Las conexiones pueden perderse antes incluso de que el servidor haya comenzado a procesar la petición, o despues de haberse procesado la petición y mientras se están transfiriendo los documentos solicitados. Una conexión perdida durante la etapa de solicitud se anota como una <<request lost connection>> (conexión perdida en petición). Una conexión perdida durante la etapa de transferencia se anota como una <<send lost connectio>> (conexión perdida en envío).

A continuación se presenta algunas de las entradas de un registro de error real:
[Thu Jun 13 13:52:00 1996] request lost connection to client 206.42.98.113
[Thu Jun 13 13:52:00 1996] send lost connection to client 206.42.98.113

Recomendación
Aunque los servidores no creen una entrada diferenciada en el registro de transferencia para las transferencias canceladas, el registro de transferencia puede dar alguna información sobre la cancelación. Las peticiones GET que se cancelan a la mitad de la transferencia aparecen con un código de estado 200 (transferencia correcta normal), pero el numero de bytes transferidos (en el ultimo campo del registro de transferencia) puede ser menor que el numero real de bytes del objeto. Sin embargo, algunas cancelaciones se producen demasiado tarde (despues de que el servidor haya terminado la transferencia, la haya anotado y esté esperando que se vacíen los buffers de salida) por lo que no se pueden detectar de esta manera.

Tiempo de Espera Excedidos
Las situaciones que provocan los fallos por tiempo de espera excedido varían mucho de un servidor a otro. No son tan frecuentes como los de conexión perdida, sino que aparecen de forma ocasional. Normalmente, los errores por tiempo de espera excedido son el resultado de la espera que debe realizar el servidor al acceder a algún dispositivo o servicio distinto al cliente que ha realizado la conexión.

Por ejemplo, un problema con una unidad de disco o un sistema de archivos montado por NFS, puede producir un error por tiempo de espera excedido. Otro ejemplo es la interrupción de la conexión física con Internet cuando el modem del lector pierde la señal de línea.

Información Útil del Registro de Error

Cuando se piensa sobre como han evolucionado los archivos de registro con el paso del tiempo, la legibilidad del registro de error para las personas, parece tener algo de sentido. Si el único archivo de registro fuera el registro de transferencia los administradores tendrían que sumergirse en megabytes de datos de registro para encontrar los errores, anotar los códigos de error y descifrar la densa información. Parece que tiene sentido disponer de otro archivo de registro que capture únicamente los errores y se anoten de tal forma que los administradores puedan solicitar los problemas que se presenten en su nodo web. Si esas fueran únicamente las diferencias entre los registros de transferencia y de error, se podría ignorar este último, por ser una duplicación inútil de la información del registro de transferencia.

Sin embargo, la información del registro de error no es simplemente la duplicación de la información que ya se encuentra en el registro de transferencia. En el registro de error existe información adicional que no se encuentra en el registro de transferencia, que puede resultar muy útil para ajustar eficazmente a un nodo web. Esto ocurre, por ejemplo, con las entradas que indican conexiones perdidas, producidas, normalmente cuando un lector cancela transferencias a mitad del proceso.

2.5.1.3 EL REGISTRO DE REMITE

El registro de remite tiene solamente dos campos de información, separados por un símbolo que se asemeja a una flecha, los campos situados a los lados de la flecha son los siguientes:

- El URL de la página que estaba mostrando el visualizador web del lector inmediatamente antes de acceder al camino y nombre de archivo.
- En camino y nombre del archivo de la página, perteneciente al árbol de documentos del nodo web, que está visualizando el lector.

Esto significa que cuando los lectores se mueven el interior de un nodo web, el lado izquierdo de la flecha es el URL de la página del nodo donde se ha pulsado el botón del ratón, y el lado derecho de la flecha es el camino/nombre de archivo de la página a la que se ha llegado después de pulsar sobre el enlace. Esta no es una información realmente útil cuando el origen y el destino se encuentran en su propio nodo, puesto que ya se dispone de dicha información en el registro de transferencia. Por ejemplo, si observamos esta entrada del registro de remite:

http://cidermill.com/info -> /info/aboutus

En el registro de transferencia se encuentra otra representación de estas líneas:

usr.az.us - - [12/Jun/1996:10:19:46 -0700] "GET /info/ http/1.0" 200 1776

usr.az.us - - *[12/Jun/1996:10:19:26 -0700] "GET /info/ aboutus http/1.0 200 2915*

En la entrada del registro de remite, se puede observar que un lector ha seguido un enlace da la página <<info>> a l página por defecto del subdirectorio llamado <<aboutus>>. Esto es todo lo que se puede extraer del registro de remite. Las dos líneas anotadas en el registro de transferencia ofrecen la misma información, pero con mucho más detalle.

Realmente las dos líneas del registro de transferencia no dicen explícitamente que el lector haya seguido un enlace de la página <<info>> a la página <<aboutus>>, aunque se puede inferir esto con total seguridad. Después de esto, la página <<aboutus>> ha sido la siguiente página solicitada después de la página <<info>>, ¿Y de qué otra manera puede el lector haber conocido la página <<aboutus>>?.

Por otro lado cuando el lector entra en la página por primera vez, el registro de tr5ansferenciamuestra realmente de donde procedía. Esto sí que es información útil. Por ejemplo, si se considera la siguiente entrada al registro de remite:

http://search.yahoo.com/bin/search?p=Apples&a=n -> /index.html

Esta entrada muestra que el lector ha realizado una búsqueda de acuerdo con algún criterio de búsqueda Yahoo. El resultado de esta búsqueda ha sido el envío de una página como la siguiente:

Figura 2 Página mostrada según criterio de búsqueda

Cuando el lector pulsa en el enlace de nuestro nodo, su visualizador envía el último URL al servidor web, de manera que pueda anotarse en el registro de remite junto con la página resultado de la búsqueda. En este caso, la búsqueda ha llevado al lector a la página /index.html, página principal del nodo.

Confidencialidad y Registro de Remite

Cuando un lector sigue un enlace de una página web a otra página web, la información de remite (el URL de la página que contiene el enlace) se codifica en los campos de la cabecera de la petición http. Sin embargo, el envío de la información de remite es una función opcional del visualizador web del lector, al menos en términos de la especificación http. Si un servidor web recibe una petición http sin información de remite, procesará la petición, pero no anotará nada en el registro de remite, porque no tiene nada que anotar.

La especificación del protocolo http recomienda fervientemente que los agentes de usuario (visualizadores web) hagan que el envío de información de remite sea una opción de los usuarios finales. Después de todo, éstos pueden desear quedar en el anonimato, y las páginas desde las que llegan a un nodo web pueden ser privadas. Desgraciadamente, los desarrolladores de visualizadores han ignorado completamente asta recomendación y no ofrecen ninguna opción de configuración para que el usuario pueda desactivar el envío de información de remite.

Por otro lado, si su interés principal son las estadísticas del nodo web, se alegrará de que los usuarios no puedan desactivar el envío de información de remite. Si la gente pudiera acceder a un nodo de forma anónima, la mayoría lo haría, por lo que no se dispondría casi de información sobre la procedencia de los lectores.

Ajuste de los Datos de Remite.

Si se registran los datos de remite, cada vez que un usuario utilice un enlace del nodo web para pasar a otra página del mismo nodo, se anotará una entrada en el registro de remite. Como ya se ha mencionado, puesto que estas entradas son esencialmente duplicaciones de los datos que se encuentran en el registro de transferencia (y no son demasiado útiles cuando el origen y destino se encuentran en el mismo nodo), se puede decidir no registrar los datos de remite del mismo nodo.

Los desarrolladores de servidores web han incorporado una forma de evitar el registro de los datos de remite de dominios o partes de dominios específicos. En los servidores Unix, se utiliza la directiva RefererIgnore para mostrar que el servidor no registró los datos de remite de dominios o de una parte del dominio. Básicamente, para no registrar la información de registro del propio dominio, basta con incluir una línea, parecida a la que se muestra a continuación, el el archivo httpd.conf:

RefererIgnore cidermill.com

Vale la pena indicar que cidermill.com es el nombre del dominio del cual no queremos entrada al registro de remite, el cual se puede sustituir por el dominio que desee.

Esto hará que el servidor sólo registre información de remite de las peticiones http que no tengan cidermill.com en la cabecera de la petición.

Como Prescindir del Registro de Remite.

La mayoría de los servidores web también permiten combinar los datos de remite con los del registro de transferencia.

Combinar los datos del registro de remite con el registro de transferencia tiene mucho sentido. Si se anota la información de remite por separado en el registro de remite, no se dispondrá de las fechas o nombres de host para hacerlo coincidir con las entradas correspondientes al registro de transferencia. Si por el contrario, se anotan los datos de remite dentro del registro de transferencia, estos aparecerán a continuación de los datos importantes del registro de transferencia.

Los datos registrados tanto en el registro de remite como en el registro de agente tienen más utilidad en el registro de transferencia. El problema es que una vez que se pasa al Formato del registro combinado, el registro de transferencia ya no tiene el Formato de registro común, y posiblemente algunos de los paquetes de software de análisis de registro puedan fallar.

2.5.1.4 EL REGISTRO DE AGENTE

Al igual que el registro de remite el registro de agente es muy sencillo. De igual modo, la información del registro de agente encaja mejor en la información del registro de transferencia con Formato de registro combinado, esto es, si se desea tener la posibilidad de enlazar los datos del agente de usuario con los accesos y visitas específicas del registro de transferencia.

Formato del Registro de Agente.

Técnicamente hablando, todo el contenido de la entrada del registro de agente de usuario es una única cadena de caracteres que envía el visualizador web remoto en uno de los campos de la cabecera de una petición http. Cada desarrollador de visualizador web define su propia cadena de caracteres cuyo formato se encuentra codificado dentro del código del programa. De todas formas, aunque las cadenas de registro de agente las definan los desarrolladores de visualizadores.
Los primeros caracteres que aparecen en una entrada del registro de agente forman una palabra, el nombre del visualizador web que utiliza el lector.

Junto con el nombre del visualizador se encuentra una barra inclinada (/) y el número de la versión del software. A continuación, a parece un espacio y otra cadena de texto. Este segundo campo está limitado con paréntesis y contienen el nombre de los sistemas operativos con el que se ejecuta el visualizador del lector, y en algunos casos información adicional acerca de la computadora.

Resumiendo, se puede decir que el formato del registro de agente es el siguiente:

<*Nombre del visualizador*>/<*versión*> (<*sistema operativo*>)

Información Útil del Registro de Agente.

Uno de los problemas principales del registro de agente es que no hay forma de ligar sus entradas con el acceso individual del registro de transferencia. No hay nombre de

computadora, ni fecha, por lo que los datos anotados en el registro de agente están completamente aislados.

Para que Sirve esta Información.

La pregunta real que hay que hacerse sobre la información de agente de usuario es: ¿Para qué sirve? Aunque las estadísticas anteriores pudieran despertar la curiosidad, su utilidad no es tan obvia para la mayoría de los administradores de sistemas o personas dedicada al marketing. En algunas ocasiones puede ocurrir que un nodo web está dirigido hacia usuarios de un visualizador o sistema operativo específico; en ese caso, si se puede realizar algún tipo de análisis útil. Pero estos casos van a constituir realmente una excepción. Generalmente, la información de agente de usuario va a ser realmente útil en los dos casos siguientes:

- Indicación de fallo, errores o anomalía de registros encontrados o creados con un visualizador o con una versión de un visualizador específico.
- Identificación y eliminación de los efectos que producen los programas spider del web y los robots que inflan y distorsionan las estadísticas de tráfico.

Otros Problemas de los Archivos de Registro.

Como ya se comentó. La información de agente de usuario queda mejor en el registro de transferencia con formato de registro combinado, puesto que el registro de agente no da pautas para ligar sus entradas con una visita particular del registro de transferencia. No hay fecha, ni nombre de computador, no hay forma de ajustar toda la información, a menos que se introduzcan los datos de remite y agente en el registro de transferencia. Por eso, si desea saber cual es el visualizador que posee un visitante, debe utilizar el Formato de registro combinado.

La otra pregunta es, ¿Por qué no se usa siempre el Formato de registro combinado? Muchos administradores de sistemas utilizan el formato de registro combinado, especialmente aquellos que no utilizan herramientas o servicios de análisis.

Hay un par de razones que explican por qué el formato de registro combinado no sirve para todo el mundo. La razón más importante es que no pueda utilizar el software o servicio de análisis de registro que prefiera. Alguna de estas herramientas y servicios solo funcionan con el registro de transferencia con Formato común de registro estricto. Pero casi todos ello son productos de libre distribución de baja calidad. La mayor parte de las mejores herramientas de análisis de registro pueden importar y

utilizar fácilmente formatos de registro variado, entre los que se encuentran distintos formatos combinados.

Además, con el formato combinado se puede desaprovechar una cantidad de disco muy significativa, al almacenar información redundante en cada uno de los accesos de una visita. Por ejemplo, pongamos que un único lector visita un nodo web y genera un total de 100 accesos. Esto significa que habrá 100 líneas más en el registro de transferencia. Cada una de esas 100 líneas tiene información útil de remite relacionada con esta visita, el URL de la página de donde procede, que se registra solo en el primer acceso al nodo web. El campo remite del resto de las entradas de los accesos tendrán el URL de la página del propio nodo web que ha seguido el lector para llegar a la nueva página, información de la que ya se dispone en la secuencia de acceso cronológico.

De la misma manera la información de agente de usuario que se anota en el registro de transferencia será redundante. Durante una visita a un nodo web, el lector solo utiliza un visualizador web. Si se combinan los datos de agente de usuario en el registro de transferencia, el servidor repetirá el nombre del visualizador del lector en cada línea del registro retransferencia.

Por ello, tiene sentido que los desarrolladores del primer servidor web decidieran colocar la información de remite y de agente de usuario en otro archivo independiente. En problema está en como lo hicieron. Aunque los servidores escriban los datos de agente en otro archivo de registro, lo siguen haciendo por cada uno de los accesos. Por tanto en el escenario descrito anteriormente, en el que una visita a generado 100 accesos, los servidores web describirán todas estas entradas en los registros de agente y remite, ahorrando muy poco (o nada) de espacio.

Otra de las desventajas del formato combinado es que cuando se combinan los datos de remite y agente de usuario en el registro de transferencia, la longitud de las líneas del registro son muy largas, haciendo muy difícil consultar el archivo de registro. El administrador del sistema tendrá que introducir el registro de transferencia en un sistema de gestión de base de datos o desarrollar filtros particularizados para eliminar la información redundante y dejarlo más legible.

2.6 ANÁLISIS DE VISITAS.

Cuando se introduce una dirección en el cuadro Location (dirección) del visualizador web o si se sigue un enlace de hipertexto para ir a un nuevo nodo, el visualizador web realiza una secuencia de pasos para contactar con el host; si tiene éxito, el visualizador y el servidor mantendrán una conversación, denominada transacción http, que esta formada por una petición http (que realiza el visualizador) y una respuesta http (que realiza el servidor).

Es muy importante tener en cuenta que cada uno de los accesos (cada una de las líneas o entradas) del registro de transferencia del servidor, se responde con una de estas transacciones http y se debe tratar individualmente. Según se vaya avanzando en el análisis de esta visita, puede dar la impresión de que, de alguna manera, el servidor sabe que la colección de acceso que se está observando está lógicamente agrupada en lo que se denomina visita. Pero el servidor no lo sabe y no tiene ninguna manera de saberlo.

Por ejemplo, la mayoría de la gente que navega por la red, lo hace desde su propia computadora monousuario. Cuando esos lectores visitan un nodo web, se puede asegurar con relativa seguridad que las trazas que se dejan en los archivos de registro sólo se refieren a ellos.

Una computadora por ejemplo, pc12.xyz.com puede ser una estación de trabajo o una computadora con 38 usuarios conectadas y navegando por la red.

Suponer que los accesos de una única computadora proceden de la misma persona puede ser erróneo. Y si se hace esta suposición y tres o cuatro de los 38 usuarios de la computadora multiusuario visitan un nodo al mismo tiempo, se obtendrá una incomprensible <<visita>> única de esa computadora, que distorsionará cualquier estadística que se extraiga de los datos del registro.

Antes de comenzar el examen de la visita, es preciso hacer una advertencia. Si se examina cuidadosamente el registro de transferencia y se extraen accesos de una computadora en particular de la manera que se hace aquí, es muy posible que la fecha de los accesos de los gráficos que aparecen en las páginas http no se encuentren tan ordenadas o de forma tan secuencial como lo hacemos aquí.

Vamos a utilizar un ejemplo encontrado en uno de los libros utilizado para el desarrollo de esta tesis: se trata de una empresa de palos de golf, esta empresa posee un servidor web y utiliza una opción de configuración llamada directiva KeepAlive, la misma que indica al servidor que no inicia un nuevo proceso para cada petición http que llegue a través de una conexión abierta.
Por ejemplo, la página principal del nodo de golf tiene unos 14 objetos además del documento HTML propiamente dicho. Sin la directiva KeepAlive, el servidor puede crear nuevos procesos para transferir cada uno de esos objetos, y hacerlo simultáneamente. Esto genera una carga innecesaria en el servidor y reducir el ancho de banda, y hace más difícil (si no imposible) determinar cuanto tiempo tardan en realizarse las transferencias. Con la directiva KeepAlive, el servidor despacha las peticiones de cada computadora secuencialmente.

Inicio del Análisis.
Para hacer el análisis de visitas utilizaremos el ejemplo del servidor de la empresa de los palos de golf y no el de la ESPOCH ya que hay algunos casos que hemos visto necesario mencionar.

Hay que mencionar que este servidor está en un sistema Unix. Para empezar se utilizó el comando more para mostrar el contenido del registro de transferencia del nodo de golf y se encontró una adecuada visita de una computadora llamada pc12.xyz.com, durante todo el análisis para referirnos a esta computadora lo haremos con el nombre Juan.

A continuación, se ejecutó el comando grep para extraer del registro de transferencia todas las entradas en las que aparecerá pc12.xyz.com y se redirigieron a un archivo llamado pc12. Estos son los mandatos:

```
% grep pc12.xyz.com > pc12
% wc –l pc12
%    80 pc12
```

Cuando Juan solicitó el primer objeto del nodo de golf eran, exactamente las 15:16:19. Cuando pidió el último objeto, eran las 15:26:46. por tanto, Juan pudo acceder a mucha información mientras se encontraba en el nodo web, estas son las cinco primeras entradas:

[01/Aug/1996:15:16:19 -0700] “GET /class/clssmisc/ msgs27233.html http/1.0” 302 -
[01/Aug/1996:15:16:20 -0700] “GET /class/clssmisc/ msgs27233.html http/1.0” 404 740
[01/Aug/1996:15:16:19 -0700] “GET /backgnd.gif html http/1.0” 200 3865
[01/Aug/1996:15:16:19 -0700] “GET /golfcrc2.gif html http/1.0” 200 5629
[01/Aug/1996:15:16:19 -0700] “GET /golfcrc.gif html http/1.0” 200 8966

Véase que la primera petición de Juan ha sido un archive específico (/class/clssmisc/ msgs27233.html) en vez del punto de inicio normal que es el archivo por defecto del directorio de documentos principal del nodo web, normalmente llamado /index.html, o simplemente, /. Este es un hecho bastante curioso: ¿cómo pudo Juan entrar en el nodo a través de ese URL? Es posible que ya hubiera estado en esa página anteriormente y que la tuviera apuntada o que siguiera un enlace desde otro nodo web.

Su primera petición devolvió un código de estado 302, lo que significa que el servidor tuvo que hacer una redirección de su petición. La segunda petición que aparece que es del mismo archivo, tuvo un código de estado de 404 (No encontrado).

En el principio el nodo de golf era sdgolf.com pero después fue cambiado por golfcircuit.com, pero en ese momento ya se encontraba registrado en ciertos motores de búsquedas. Los posibles nuevos lectores que trataran de entrar en sdgolf.com a través de ese enlace encontrado por el motor de búsqueda, podrían pensar que habrían perdido la capacidad de conexión por parte de los nodos de la red.

Por ello. En vez de cambiar directamente el nombre del dominio, se añadió el nuevo, manteniendo el original, y se cambiaron todos sus enlaces de hipertexto para que apuntaran al nuevo dominio. Se configuró el servidor web para redirigir todas las peticiones realizadas con el nombre de dominio antiguo al nuevo. Esta es una característica estándar de casi todo servidor web. Así, todas las peticiones entrantes de documentos del dominio de sdgolf.com se redirigen automáticamente al dominio golfcircuit.com. Por ejemplo, las peticiones de:

http://www.sdgolf.com

Se redirigen automáticamente a:

http://www.golfcircuit.com

y

http://www.golfcircuit.com/foo/bar

Además para configurar el servidor para redirigir todas las peticiones al nuevo URL, se hizo que ambos nodos compartieran los archivos de registro. Por esta razón, en el registro de transferencia aparecen ambas entradas, tanto la primera redirección como la segunda petición, aparentemente idéntica. La primera línea es la redirección desde el antiguo dominio y la segunda es la petición redirigida al nuevo dominio.

¿Por qué falla la segunda petición con el código 404? ¿Qué significa esos 740 bytes transferidos como resultado de la petición? Una rápida comprobación en el directorio indicado revela que la petición ha fallado porque el archivo no existe. Este directorio almacena anuncios clasificados que los propios lectores introducen a través de un formulario HTML y un programa CGI. Cuando venden una mercadería o encuentran lo que están buscando, ellos mismos retiran el anuncio. Es posible que Juan viera un anuncio de un fantástico conjunto de palos de golf y decidiera apuntarlo hasta que pudiera encontrar una manera de convencer a su mujer…y se ve que le llevó demasiado tiempo.

Sin embargo, no es necesario especular tanto. Veamos si este archivo aparece en el registro de remite. Cuando se hace el grep en el registro de remite con msgs27233.html, se encontró esta línea:

http://www.excite.com/search.gw?search=GOLF, HANDICAP ->
/class/clssmisc/msgs27233.html

Esto lo explica todo, Juan introdujo algún criterio de búsqueda en un motor de búsqueda y se le presentó una página que tenía un enlace a un viejo anuncio clasificado del nodo de golf. Para confirmarlo, podemos ir al mismo motor de búsqueda e introducir el mismo criterio de búsqueda y ver que se muestra.

Veamos qué sabemos por ahora de la visita de Juan. Entró en el motor de búsqueda, realizó una búsqueda con la palabra golf, que generó una página resultado en la qué aparece un viejo anuncio de este nodo de golf.

Entonces a Juan le pareció lo suficientemente interesante para pulsar sobre el enlace. El servidor web interceptó su petición de la página del antiguo dominio sdgolf.com y la redirigió a golfcircuit.com. Entonces, el servidor trató de procesar la petición redirigida y se encontró con que el archivo ya no existe, por lo que devolvió un código 404 (no encontrado). Pero, ¿por qué transfirió 740 bytes de datos de una página que no existía?.

Otra de las opciones de ejecución disponible en muchos servidores web permite especificar un documento de error por defecto, que se envía a los clientes cuando se produce alguno de los errores previamente indicados. Esto se aprovechó en el nodo golfcircuit.com. Con la directiva ErrorDocument del servidor, se le indicó a este que enviara el archivo /notfound.html cuando se produjesen condiciones de error con código 404. Así se evita que el cliente se enfade, o por lo menos, se consigue apaciguar un poco su ánimo.

El archivo /notfound.html contiene exactamente 740 bytes de código HTML. También incluye tres imágenes: una imagen de fondo, la imagen del titulo y la barra de botones de la parte inferior de la página. Estos explica los siguientes tres accesos de las primeras cinco líneas que se esta examinando.

Después de leer la página del documento de error, Juan pulsó en el enlace <<Pulse aquí para continuar>>, lo que le llevó directamente a la página principal (el archivo /index.html, simplemente aparece como /).

Ahora vamos a centrarnos en este acceso y todos los siguientes desde el momento en que se solicita otro archivo .html. Estos son los enlaces:

[01/Aug/1996:15:16:52 -0700] "GET / http/1.0" 200 5047
[01/Aug/1996:15:16:54 -0700] "GET /art/backgnd.gif http/1.0" 200 3865
[01/Aug/1996:15:16:52 -0700] "GET /golfcirc1.gif http/1.0" 200 3177
[01/Aug/1996:15:16:52 -0700] "GET /header.gif http/1.0" 200 4961
[01/Aug/1996:15:16:52 -0700] "GET /leader.gif http/1.0" 200 1285
[01/Aug/1996:15:16:52 -0700] "GET /on_tour/pga/british/logo.gif http/1.0" 200 2827
[01/Aug/1996:15:16:52 -0700] "GET / on_tour/pga/cvs/logo.gif http/1.0" 200 3765
[01/Aug/1996:15:16:52 -0700] "GET /on_tour/pga/british/open.gif http/1.0" 200 4445
[01/Aug/1996:15:16:52 -0700] "GET /circdir.gif http/1.0" 200 3621
[01/Aug/1996:15:16:52 -0700] "GET /new.gif http/1.0" 200 116
[01/Aug/1996:15:16:52 -0700] "GET /directory/circnet.gif http/1.0" 200 3446

[01/Aug/1996:15:16:52 -0700] "GET /greendot.gif http/1.0" 200 326
[01/Aug/1996:15:16:52 -0700] "GET /Netscape.gif http/1.0" 200 1135
[01/Aug/1996:15:16:52 -0700] "GET /Banners.class http/1.0" 200 23912
[01/Aug/1996:15:16:52 -0700] "GET /BannersMsg.class http/1.0" 200 1376

Estos 15 accesos son el resultado de la carga de la página principal. La propia página (el archivo .html) contiene 5047 bytes de código HTML y hace referencia a unas cuantas imágenes e incluso a un par de applets Java.

No parece que halla ningún error o algún otro misterio sin resolver, aunque si es posible sacar más interesantes conclusiones de este segmento de registro de transferencia. En primer lugar, puede observarse que todos los accesos han tenido un código 200 (Ok). Eso es bueno, ya que no habrá ningún enlace interrumpido ni faltará ninguna imagen.

Si se observa con detenimiento esta página, con todos sus gráficos y applets Java, puede decirse que quizá sea demasiado grande.

Si se suman los bytes transferidos en cada uno de los accesos, sale un total de 63.304 bytes para una página. Si se hace la diferencia entre la fecha inicial y final, se puede afirmar que Juan tardó 34 segundos en realizar la transferencia de la página completa (realmente, no se está incluyendo el último applet Java, ya que la fecha indica cuando se inicia la transferencia y no cuando se termina de transferir el archivo).

Si realmente se deseara estimar con más precisión el tiempo de transferencia, tendría que hacerse algo como lo siguiente: en los 34 segundos anteriores al inicio de la última transferencia (el objeto /BannersMsg.class), el servidor ha transferido correctamente 61.928 bytes. Indicándolo matemáticamente:

$$\frac{61.928 bytes}{34 segundos} = 1.821 bytes / segundos$$

Por tanto, el tiempo que se habrá tardado en transferir el último objeto (el objeto /BannersMsg.class) sería:

$$\frac{1.376 bytes}{1.821 bytes / segundos} = 0.76 segundos$$

Dificultad en el Seguimiento de una Visita.

Pongámonos en el papel de un desarrollador de software. Estamos desarrollando software de análisis de visitas y reaumentas de visitas en vez de simples acceso. Para comenzar, codificaremos el software de manera que se hicieran las mismas suposiciones realizadas anteriormente: los accesos que se han originado en el mismo host en un intervalo de tiempo razonable constituye una visita; cuando no se han recibido accesos de una host después de pasados 30 minutos, se supondrá que el lector se ha marchado y que la visita ha terminado en el último acceso registrado.

Con estas suposiciones, ¿Bajo qué circunstancias serían falsas o erróneas las estadísticas generadas?

En primer lugar, se deben considerar las computadoras multiusuarios. Algunos usuarios utilizan terminales de texto (no gráficos) y trabajan con la computadora desde la línea de mandatos; algunas de estas personas disfrutan realmente con ello y prefieren navegar por la red con un visualizador basados en caracteres como Lynx. Cualquier sistema Unix medianamente cargado podría tener un par de docenas o quizás más usuarios. Puesto que todos ello comparten el mismo host, todos ello envían el mismo nombre de host hacia los archivos de registros de los nodos que visitan. Si todos ello visitan el mismo nodo web en días distintos, no hay forma de saber si se trata de una persona que visita el nodo en varia ocasiones o diferentes usuarios que realzan una única visita.

Como se puede ver, aunque hay muchísimas PC, los sistemas de computadores multiusuarios los utiliza el suficiente número de gente como para hacer que las estadísticas de tráfico se desbaraten completamente.

¿Qué suceden con aquellas personas que mientras están consultando un nodo web, sienten la necesidad de ir a buscar algo? Puede tardar 45 minutos y continuar consultando el nodo web, aunque el software habrá dado por terminada la visita pasados los 30 minutos de inactividad.

Cuando se pulsa sobre uno de los enlaces que siguen visibles en la pantalla de un monitor, el software lo tomará como el inicio de una nueva visita.

Quizá todavía peor que distorsionar el número de visitas es que el software anota esta nueva visita como si comenzara en un lugar extraño, lo que indicará una trayectoria

errática por el nodo, puesto que todas las páginas del nodo se encontraran en la memoria caché del visualizador y no se registrará ningún nuevo acceso. Esto quebrantará totalmente las estadísticas, puesto que afectará el número medio de páginas consultadas por visita y a la duración media de consultas de las páginas.

Este ejemplo supone que el lector ha dejado abierta una conexión con Internet mientras estuvo fuera o dispone de un nombre de computadora y dirección IP fijo. ¿Qué pasaría si fuera una de las miles de personas a las que se asigna la dirección IP de forma dinámica cuando estables su conexión con su proveedor de servicios? Con casi completa seguridad cuando volviera a conectarse, obtendría una dirección IP totalmente diferente y cuado volviera a conectarse se lo registrara como un visitante nuevo.

***Importancia de las Visitas*.**

Probablemente haya asimilado ya la idea de que conocer los accesos de forma individual no es suficiente, aunque no es posible de que todavía no comprenda completamente por qué aquí (y en toda la comunidad web) se está tratando de encontrar una forma de extraer la información de los acceso y utilizarla para las visitas, e incluso quizá para los visitantes. En este preciso momento, seguramente esté pensando que está a punto de descubrirlo, antes de responder la pregunto de por qué todo el mundo está cambiando la perspectiva del análisis de registro de acceso de visitas, examinemos cuales son las estadísticas que ofrecen actualmente los paquetes de análisis de registros.

Generalmente se puede decir que las estadísticas entran en dos categorías:

- Estadísticas basadas directamente en los accesos
- Estadísticas derivadas del número de visitas reales.

Se estudiaran primero las estadísticas que pueden extraerse directamente de los accesos de los registros de transferencia, se disponga o no de una forma para contar el número de visitas reales, estas estadísticas incluyen los siguientes puntos:

- Número total de peticiones
- Número total de bytes transferidos
- Número de peticiones por cada página, gráficos y archivos

- Documentos más solicitados
- Archivos más solicitados
- Formularios y scripts más utilizados
- Documentos más solicitados por directorio
- Número medio de peticiones por día o por hora
- Número medio de byte transferidos por hora
- Número medio de accesos en días laborables
- Número medio de accesos en fines de semana

Estas son estadísticas útiles, aunque dejan algo que desear. Ya sabemos que cada acceso tiene una cierta información sobre el host que realiza la petición, y que se puede extraer bien aplicando las suposiciones generales que se pueden hacer acerca del tiempo de organización (a través del nombre del dominio del host) o bien buscando los nombres de las empresas e información geográfica en una base de datos. Pero, ¿Qué validez tiene todo esto si se tiene encuenta que las estadísticas están basadas en accesos?

No tiene ningún sentido utilizar estas estadísticas como base para obtener otras estadísticas que representen, por ejemplo, el porcentaje de acceso que proceden de Colombia o porcentaje de acceso de una empresa en particular. El número puede ser real, pero ¿Qué significa? Puesto que no se puede contar una visita con facilidad, no se puede tomar el número medio de acceso por visita y dividirlo entre el número de accesos que proceden de Colombia para estimar el número de visitas de Colombia.

El seguimiento de visitas es de vital importancia si se quiere realizar un análisis profundo del nodo web. Cuanto más fiable se realice el seguimiento de visitas, se podrán obtener más estadísticas, con mejor y más interesente información. Estas son algunas de ellas:

- Número de visitas
- Número medio de peticiones por visitas
- Duración media de una visita
- Visitas de organizaciones (las organizaciones más activas)
- Visitas por tipo de organización
- Visitas por país (los países más activos)
- Páginas consultadas con mayor frecuencia al iniciar la visita

- Mayores duraciones por páginas
- Páginas consultadas con mayor frecuencia al salir de la visita
- Número medio de visitas por día/hora
- Regiones geográficas
- Ciudades principales
- Organizaciones con más remites
- Lista de URL con más remites
- Visualizadores principales
- Sistemas Operativos de usuarios más frecuentes
- Número medio de usuarios en días laborables
- Número medio de usuario en fines de semanas
- Día de la semana con mayor actividad
- Día con mayor actividad de todos
- Número de acceso en el día más activo (con mayor número de visitas)

Estas estadísticas son totalmente necesarias y sólo pueden alcanzar su significado real cuando están basadas en visitas reales y no en accesos. Esta es la razón por la cual el seguimiento de visitas es de importancia vital.

2.7. ANATOMÍA DE LAS PETICIONES Y RESPUESTAS HTTP.

Cuando un visualizador web se prepara para iniciar una petición a un servidor web, lo primero que hace es recoger información sobre sí mismo, sobre la petición que ha solicitado el lector al escribir un URL o al pulsar sobre un enlace, y sobre la computadora en que se está ejecutando. El visualizador escribe toda esta información en una especie de pequeño paquete llamado petición http. Una petición http se puede concebir como un informe escrito con el siguiente contenido:

A: www.bluebikes.com
De: win22.xyz.com
Asunto. Petición de página
Visualizador: Mozilla/3.2 (Windows 95)
Remite: http://win22.xyz.com/coollinks.html
Petición: GET /aboutus.html http/1.0

Esta puede ser una forma excesivamente simple de verlo, pero recoge perfectamente el objetivo y parte de la estructura real de una petición http. Llamaremos cabecera de mensaje a las seis primeras líneas de este mensaje. Es preciso observar que en el

cuerpo del mensaje no hay ninguna información adicional que no se encuentre ya, incluso mejor detallada, en las cabeceras del mensaje. Por ello, y puesto que no hay equivalente del cuerpo del mensaje en las peticiones http, se va abandonar. Además, tampoco hay cabecera Asunto, ya que esto se describe en la cabecera petición.
A continuación, se va a ver un ejemplo muy parecido a la realidad: el visualizador primero establece una conexión directa al puerto 80 de la computadora donde reside el servidor (en este caso www.bluebikes.com). En contestación a la conexión original, ambas computadoras se identifican entre sí, por lo que no son necesarias las cabeceras A y De. Después, el visualizador que realiza la petición envía la siguiente:

GET /aboutus.html http/1.0
If-modified-since: Fri, 6 Dec 1996 18:42:29 GMT
Referer: Http://win22.xyz.com/coollinks.html
User-Agent: Mozilla/3.2 (Windows 95)

En conjunto, esto es lo que se llama una petición http. La primera línea se llama línea de petición. Cada una de las líneas que siguen a continuación de la primera se denomina genéricamente cabecera de la petición http. En este ejemplo, se ha añadido la cabecera If-modified-since. Si un visualizador descubre que el documento solicitado se encuentra en su caché, puede añadir la cabecera If-modified-since a la petición para convertirla en una petición GET condicional, de manera que el servidor sólo enviará el documento si éste ha sido modificado después de la hora y fecha especificada. Si el documento no ha sido modificado, el servidor devolverá un código de estado 304 (no se ha modificado) y el visualizador utilizará la copia de su caché local.

La cabecera If-modified-since añade una funcionalidad muy interesante tanto para el cliente como para el servidor. Para el lector que utiliza el visualizador cliente, las páginas se cargan mucho más rápidamente como lo hacen desde la caché del disco local que cuando la página o los gráficos tienen que llegar a través de la línea telefónica. Para el servidor, significa transferir menos bytes, lo que produce un ahorro de ancho de banda.

Tanto para las peticiones como para las respuestas, no hay un número fijo de cabecera obligatoria. La petición anterior funcionaría correctamente (desde el punto de vista del lector) si se enviará solamente la primera línea. Sólo es realmente necesaria

la línea de petición. Las líneas adicionales (las líneas de cabecera) simplemente añaden funcionalidades para el cliente, el servidor o para ambos, las cabeceras Refer y User-Agent únicamente sirve para que el servidor web pueda registrar esa información en sus archivos de registro respectivamente (o en el registro de transferencia, en caso de utilizar formato combinado).

Ahora, veamos la respuesta que debería enviar el servidor web al visualizador. Al igual que la petición, la respuesta estará formada por unas única línea, llamada línea de estado, más un cierto número de líneas adicionales (opcionales) llamadas cabeceras de respuesta y un número variable de líneas llamas cabeceras de entidad. Este es un ejemplo de respuesta:

http/1.0 200 Ok
Server: CERN/3.0 libwww/2.17
Content-type: text/html
Content-length: 2427
<Cuerpo de la Entidad>

Esta respuesta http está formada por una línea de estado (la versión del protocolo y el estado del resultado de la petición) seguida por una cabecera de respuesta (la cabecera Server), dos cabeceras de entidad (Content-type y Content-length) y una cabecera de bytes (representadas con <Cuerpo de la entidad>) que es el contenido del documento solicitado. La cabecera Server al nodo web en su totalidad y no tiene nada que ver con el objeto que se va a transferir al visualizador. La cabecera Content-type y Content-length se refieren directamente a la entidad que viene a continuación de las cabeceras (el texto del documento html en sí). Las cabeceras indican al visualizador del lector que el documento es un archivo de texto html y que contiene 2.427 bytes.

Generalmente, las cabeceras de respuesta pueden estar relacionadas con cualquier motivo aunque no está directamente relacionado con la entidad solicitada. Las tres primeras cabeceras principales se describe en la siguiente tabla:

Cabecera	Utilización
Location	Cuando un visualizador encuentra esta cabecera, se hace una redirección a otro URL.
Server	Especifica el nombre del software del servidor
WWW-authenticate	Indica que será necesario realizar una identificación antes de que el usuario pueda recibir la entidad solicitada.

Tabla 8 Cabeceras de Respuestas

Las cabeceras de entidad están relacionadas, o describen a la entidad solicitada que va ha enviarse. Hay más cabecera de entidad que de respuesta. Las cabeceras de entidad más frecuentas se muestran en la siguiente tabla:

Cabecera	Utilización
Allow	Especifica los métodos (por ejemplo: GET, HEAD o POST).
Content-encoding	Si la entidad ha sido codificada (comprimida o cifrada), esta cabecera receptora puede determinar cómo descodificarla para obtener su formato inicial.
Content-length	Tamaño (número de bytes) que contiene el objeto.
Expires	Adjunta una fecha a la entidad, pasada la cual, cualquier agente de usuario debe considerar la entidad como no válida.
Last-modified	Indica la fecha y la hora en la que se ha modificado la entidad por última vez.

Tabla 9 Cabeceras de Entidad

Aunque hay muchas cabeceras de entidad, es bastante raro que el servidor envíe más cabeceras aparte de las del formato del contenido y su longitud. De hecho, muchos visualizadores no suelen reconocer la mayoría de las cabeceras.

Como se puede observar, en la respuesta http del ejemplo anterior, no se delimitan claramente las cabeceras de respuesta y las de entidades. Así es como funciona en la práctica. No importa quien genera la cabecera, sea un servidor internamente a si es

un programa CGI (interfaz de pasarela común). Si el visualizador receptor está programado para reconocer las cabeceras y hacer algo con ella, lo hará.

Por ejemplo, supongamos que un programa CGI genera esta salida:

Content-type: text/html
Location: http://www.bluebikes.com/index2.html

El visualizador recibirá las cabeceras y se redirigirá al URL designado en la cabecera Location. El visualizador no muestra ninguna de estas líneas, porque las reconoce como líneas de cabeceras. Lo único que hace es realizar una nueva petición, de forma silenciosa, al documento que le ha sido indicado.

Según las estrictas definiciones dadas anteriormente para las cabeceras Content-type pertenece a la categoría de entidad, mientras que la cabecera Location es una cabecera de respuesta. Este es un buen ejemplo que explica porque la separación entre las cabeceras de respuesta y entidad está tan difundida y porqué todas estas cabeceras que se envían desde un servidor a un cliente se suelen conocer genéricamente como cabeceras de respuesta.

2.8. AFINAMIENTO DEL SERVIDOR PARA MEJORAR SU EFICACIA.

Si se desea mejorar la eficiencia del nodo web, una de las primeras cosas en que se debe pensar es en el objetivo del nodo. Hay, básicamente, tres categorías principales:

- *Establecer la presencia de una organización en la red:* Esto puede comprender las tareas de informar al público o a los posibles clientes acerca de la empresa, dar a conocer sus productos y servicios, y atender a la base de clientes actuales a través de un servicio al cliente y de asistencia técnica basados en web. Aunque casi todas las empresas pueden vender sus productos y servicios a través del nodo web, si el objetivo principal del nodo es vender, probablemente se encuentre dentro de esta categoría.
- *Vender producto desde el nodo web:* Este web puede, además, ofrecer servicios al cliente y asistencia técnica. Se debe hacer un énfasis especial: si el objetivo principal es vender productos, los objetivos para ajustar el nodo será diferente.

- *Vender publicidad en el nodo web:* El primer objetivo de este tipo de nodos es proporcionar unos contenidos impactantes y novedosos para atraer al mayor número posibles de visitantes. El objetivo es maximizar la exposición de la publicidad que encargan los anuncios.

Cada una de estas categorías tiene sus propios requisitos para el diseño, contenido y seguimiento de nodos. No es casual que la importancia del seguimiento del tráfico del nodo y el ajuste de su contenido, se incremente de forma exponencial entre estas categorías, de la primera a la tercera.

2.9 CONTROL DE ANCHO DE BANDA

Una tarea cada vez más solicitada a los administradores es el control del ancho de banda. El Ancho de Banda en sí mismo es una función del tamaño y el tiempo: por ejemplo, la velocidad la medimos en metros por segundo. En el mundo de las comunicaciones, medimos bits, bytes o algún múltiplo por segundo. De esta forma, tenemos que en un vínculo de 512kbit/s logramos una velocidad o tasa de transferencia de 64 kilobytes por segundo, ya que 8 bit = 1 byte y por lo tanto 512 / 8 = 64. De aquí que las limitaciones de ancho de banda se realicen intercalando esperas en la transmisión / recepción de datos.

Si suponemos un escenario típico, tendremos direcciones IP (computadoras, o dispositivos) que intentan acceder a recursos IP de otras redes, pasando a través de un gateway. Esto se puede aplicar tanto a una configuración NAT clásica: direcciones IP privadas accediendo a Internet a través de un gateway GNU+Linux con dos o más interfaces de red.
Como sea, logramos que todas las solicitudes hacia Internet pasen a través de nuestro gateway por lo que podremos controlar, no solo que protocolos o combinaciones de puertos o IP de origen o destino permitir, sino también el ancho de banda permitido.

Desde hace algún tiempo se ha estado intentado ecualizar el ancho de banda a una conexión, con el fin de poder garantizar que siempre se dispone de un ancho de banda mínimo para un determinado servicio o máquina de una red.
Para entender la utilidad de esto supongamos que tenemos una red como la siguiente:

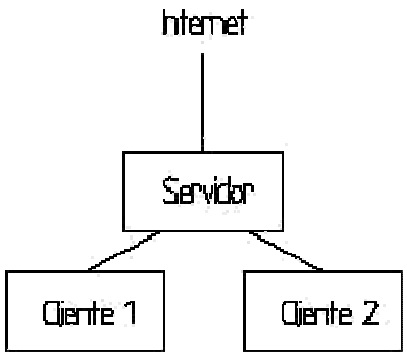

Figura 3 Gráfica de una red

es muy habitual encontrarse con el hecho de que el *cliente 1* comienza a utilizar ancho de banda de forma masiva y continua (ftp, p2p, etc...); de manera que si en un momento posterior el *cliente 2* intenta utilizar la conexión para cosas más livianas (navegar), este descubre que la conexión esta copada por el primero y todo el acceso le va muy lento.

Se desea encontrar un método para poder evitar esta u otras situaciones en las cuales el ancho de banda es utilizado de una manera poco justa entre los usuarios (servicios) o como mínimo de una manera no adecuada. Entre otras cosas se pretende explicar la forma de conseguir lo siguiente:

- Asegurar que no haya en la red usuarios (o servicios) que utilizan el ancho de banda ocasionando un acceso precario para los otros
- Establece prioridades entre las máquinas cliente de nuestra red (si esto fuese necesario)
- Garantizar un ancho de banda mínimo para cada servicio o grupo de servicios (ftp, telnet, ssh, www, etc ...)

Se dará a conocer algunos de los servicios QoS (quality of service) que están disponibles en el núcleo de Linux; estos servicios se pueden utilizar de muchas formas y combinar entre ellos para solucionar problemas muy complejos.

2.9.1 ALGORITMOS PARA EL CONTROL DE ANCHO DE BANDA

Existen muchos métodos para controlar el ancho de banda, cada uno se ajusta a las necesidades de quien los necesite. A continuación se describirá a algunos de estos:

CBQ

CBQ utiliza un algoritmo basado en la estimación del intervalo de tiempo trascurrido entre dos peticiones consecutivas del hardware para el envió de datos.

CBQ es la más compleja, menos entendida y más empírica de todas las disciplinas de colas. Esto obedece básicamente al hecho de que, además de ser una disciplina de colas con clase, CBQ ofrece la capacidad de regular el ancho de banda (shaping), siendo en este terreno donde surgen sus mayores dificultades.

Supongamos que se intenta reducir el ancho de banda de una conexión de red de 10 Mbit/seg a 1 Mbit/seg. Para lograrlo tendremos que conseguir que el enlace esté inactivo el 90 % del tiempo. Sin embargo, medir con exactitud ese porcentaje entraña mucha dificultad. Así, se ha comprobado que no siempre se consiguen buenas aproximaciones de este modo, e incluso a veces los resultados son totalmente equívocos.

No obstante, para la mayoría de las situaciones, este algoritmo trabaja de forma adecuada cumpliendo perfectamente con nuestras necesidades.

Parámetros CBQ para ajustar la regulación del ancho de banda.

CBQ trabaja intentando que el enlace esté inactivo durante un porcentaje de tiempo que asegure que se regula hasta conseguir el ancho de banda deseado.
También hemos hecho referencia a la complejidad del algoritmo y su falta de exactitud en determinadas situaciones. De hecho, el número de parámetros que se utilizan para esta tarea es elevado y a veces no se sabe cuál es exactamente su función, de ahí que en la mayoría de ocasiones debe ser la propia experiencia la que enseñe como jugar con estos parámetros.
Algunos de los más importantes son:

- **avpkt**. Tamaño medio del paquete medido en bytes
- **bandwidth**. Ancho de banda del dispositivo físico. Se necesita para calcular el tiempo muerto entre petición y petición.
- **mpu**. Tamaño mínimo del paquete. Es necesario porque incluso un paquete de cero bytes de datos da lugar a una trama Ethernet de un tamaño mínimo distinto de cero.
- **rate**. Ancho de banda regulado con el que queremos que funcione nuestra disciplina de colas.

Parámetros CBQ para definir la posibilidad de prestar y pedir prestado ancho de banda entre las distintas clases.

Manteniendo siempre la limitación global en el ancho de banda establecido para la disciplina de colas CBQ, existe la posibilidad de que entre las clases se presten ancho de banda en el caso que sea posible. Los parámetros de que se dispone para ellos son:

- isolated / sharing. Una clase configurada con el parámetro isolated no prestará nunca ancho de banda a sus hermanas. El comportamiento contrario viene establecido por el parámetro sharing. Por defecto, si no se indica lo contrario se supondrá que el sharing está activo.
- bounded / borrow. Una clase configurada con el parámetro bounded no intentará pedir prestado ancho de banda a ninguna de sus hermanas. El comportamiento contrario viene establecido por el parámetro borrow. Por defecto, si no se indica nada, se supondrá que el borrow está activo.

Básicamente CBQ es muy apropiada para casos en los que se dispone de un ancho de banda fijo que se quiera dividir en varios propósitos, dando a cada uno de ellos un ancho de banda garantizado, y con la posibilidad de prestar ancho de banda entre ellos.

Control de ancho de banda con CBQ-INIT

Este script sirve para simplificar la disposición y la gerenciamiento del control de tráfico de manera relativamente simple usando un sistema basado en CBQ en Linux. El acceso a las características avanzadas de red del núcleo de Linux es proporcionado por la utilidades "ip" y "tc" del paquete de iproute2 de A. Dado que estas utilidades sirven específicamente para traducir deseos del usuario a comandos de RTNETLINK, su interfaz es algo espartana, intolerante y requiere la escritura de largas cadenas de comandos con parámetros. Esta tediosa tarea de escritura de comandos es lo que este script procura reducir:).

El manejo de las facilidades avanzadas de una red en Linux es bastante flexible y este script trae algunas de estas características a los usuarios de Linux. Por supuesto, hay una relación inversamente proporcional entre la simplicidad y la flexibilidad y usted puede pensar que la flexibilidad sufrió demasiado en favor de la simplicidad y que no cubre todas sus necesidades, entonces es hora de hacer frente al interfaz "ip" y "tc".

Configuración de CBQ-init

Cada clase de tráfico se debe describir con un archivo en $CBQ_PATH en el directorio (/etc/sysconfig/cbq por defecto), un archivo por clase. Los nombres del archivo de la configuración deben obedecer un formato obligatorio: cbq-<clsid>.<name> donde <clsid> es un numero hexadecimal de dos bytes dentro del rango <0002-FFFF> (que de hecho es una identificación de la clase de CBQ) y <name> es el nombre de la clase, cualquier cosa que le ayude a distinguir el archivo de configuración. Para una cantidad pequeña de clases que es posiblemente la mayoria de los casos deje que el <clsid> refleje el ancho de banda de la clase.

Parámetros de dispositivo

Parámetro:

DEVICE=<ifname>,<bandwidth>[,<weight>]

Ejemplo:

DEVICE=eth0, 10Mbit, 1Mbit

Descripción:

Cuando usted tiene más de una clase en un interfaz, basta con especificar ancho de banda y peso solamente una vez, por lo tanto en otros archivos usted necesita solamente especificar DEVICE=ifname.

Parámetros de Clases

Parámetro:

RATE=<speed>

Ejemplo:

RATE=5Mbit

Descripción:

Ancho de banda asignado a la clase. El tráfico que pasa a través de la clase se regula para adecuarse a la medida asignada. Usted puede utilizar Kbit, Mbit o los BPS, Kbps y Mbps son sufijos. Si usted no especifica ninguna unidad, se utilizan bits/sec. También observe que los "BPS" significan "octetos por segundo", no los bits.

Parámetro:

WEIGHT=<speed>

Ejemplo:

WEIGHT=500Kbit

Descripción:

Parámetro de tunning que debe ser proporcional a RATE. En general, WEIGHT ~= RATE / 10.

Parámetro:

PRIO=<1-8>

Ejemplo:

PRIO=5

Descripción:

Prioridad del tráfico de la clase. Cuanto más alto es el número, menor es la prioridad. La prioridad de 5 el valor por defecto.

Parámetro:

PARENT=<clsid>

Ejemplo:

PARENT=1280

Descripción:

Especifica la identificación de la clase con el padre a la cual usted desea que esta clase este vinculada puede ser que usted desee utilizar LEAF=none para la clase del padre según lo mencionado abajo. Usando este parámetro cuidadosamente en los archivos de configuración, es posible crear las estructuras jerárquicas simples de las clases de CBQ. El orden es importante debe construir clases padre antes de sus hijos.

Parámetro:

LEAF=none|tbf|sfq

Ejemplo:

LEAF=none

Descripción:

Dice al script que relacione la disciplina de encolamiento que especifico a la clase de CBQ. Por el defecto, es utilizado TBF.

Observe que adjuntando TBF a la clase de CBQ controla el tráfico para conformar los parámetros de TBF y evita que la clase pida prestada ancho de banda de su padre incluso si usted HA LIMITADO el sistema a "no". Para permitir que la clase pida prestado el bandwith, usted a debe fijar LEAF "none" o "sfq".

Si usted desea asegurar compartir del recurso de ancho de banda entre varios anfitriones en la misma clase, puede ser que desee especificar LEAF=sfq para adjuntar SFQ como disciplina que de encolamiento de esa clase.

Parámetro:

BOUNDED=yes|no

Ejemplo:

BOUNDED=yes

Descripción:

Si se seta a "yes", la clase no se permite pedir prestado ancho de banda de la clase del padre en la situación de sobrepasar el limite asignado. Si se setea a "no", la clase permite pedir prestado ancho de banda de su padre.

Nota: No se olvide de fijar la LEAF a "none" o a "sfq", si no la clase tendrá TBF unida a sí mismo y no podrá pedir prestado el ancho de banda si usar de su padre.

Parámetro:

ISOLATED=yes|no

Ejemplo:

ISOLATED=yes

Descripción:

Si se setea a "sí", la clase no presta ancho de banda no usada a sus hijos.

Parámetros de TBF qdisc

Parámetro:

BUFFER=<bytes>[/<bytes>]

Ejemplo:

BUFFER=10Kb/8

Descripción:

Este parámetro controla la profundidad del token bucket. En otras palabras representa el tamaño máximo de burst que la clase puede enviar. La parte opcional del parámetro se utiliza para determinar la longitud de los intervalos en el tamaño de los paquetes, para los cuales se guardan los tiempos de la transmisión.

Parámetro:

LIMIT=<bytes>

Ejemplo:

LIMIT= 15Kb

Descripción:

Este parámetro determina la longitud máxima de la reserva. Si la coleta contiene más datos que los especificados por LIMIT, se descartan los paquetes nuevos que llegan. La longitud de la reserva determina la latencia de la cola en caso de congestión.

Parámetro:

PEAK=<speed>

Ejemplo:

Descripción:

Tarifa máxima para el tráfico a censurado plazo de la explosión. Esto permite que usted controle la tarifa máxima absoluta que la clase puede enviar, porque solo TBF

que permite 256Kbit/s por supuesto permitiría el índice de 512Kbit para la mitad de un segundo o un 1Mbit para un cuarto de segundo.

Parámetro:

MTU=<bytes>

Ejemplo:

MTU=1500

Descripción:

Número máximo de octetos que se pueden enviar inmediatamente sobre el medio físico. Se requiere este parámetro cuando usted especifica parámetro PEAK. Por lo general se omite el MTU de Ethernet, pero, para otros tipos de medios puede ser que desee cambiarlo.

Nota: Fijar TBF como qdisc de la hoja evitará con eficacia que la clase pida prestado ancho de banda de la clase padre, porque incluso si la clase permite que más tráfico pase a través de ella, este se adapta para conformarse con TBF.

Pará metros de SFQ qdisc

Pará metro:

QUANTUM=<bytes>

Ejemplo:

Descripción:

Este parámetro no se debe fijar más bajo que el MTU, para Ethernet es 1500b, o (con la cabecera MAC) 1514b que es el valor usado en los ejemplos de Alexey Kuznetsov.

Parámetro:

PERTURB=<seconds>

Ejemplo:

Descripción:

Período de la perturbación de la función hash. Si no se setea, la reconfiguración del hash nunca ocurrirá, que es lo que usted probablemente no desea. El valor prefijado de 10 segundos es uno bueno.

Parámetros de filtrado

Parámetro:

RULE=[[saddr[/prefix]][:port[/mask]],][daddr[/prefix]][:port[/mask]]

Ejemplo:

RULE=10.1.1.0/24:80

Descripción:

Estos parámetros levantan las reglas de filtrado "u32" que deriva el tráfico por cada una de las clases. Usted puede utilizar campos múltiples de RULE por cada

configuración. La máscara de puertos opcional se debe ser utilizada solamente por usuarios experimentados que entienden cómo funciona el filtro u32.

Pará metro:

REALM=[srealm,][drealm]

Ejemplo:

REALM=russia, internet

Descripción:

Estos parámetros levantan las reglas del filtro "route" que clasifican tráfico según reinos del paquete origen/destino. Para mas información sobre reinos, vea el Alexey Kuznetsov's IP Command Reference. Este script no define ningún reino, él solo construyen los comandos de "filtro de tc" para usted si necesita clasificar tráfico de esta manera. El reino es un número decimal o una secuencia que se refiere a a/etc/iproute2/rt_realms de la entrada.

Parámetro:

MARK=<mark>

Ejemplo:

Descripción:

Estos parámetros levantan las reglas del filtro del "fw" para cada una de las clases de acuerdo a "mark" del cortafuego. La marca del paquete es un número decimal que etiqueta cada paquete que cumple una regla si una regla del firewall así lo dicta. Usted puede utilizar campos múltiples de la MARK por cada configuración.

Nota: Las reglas para diversos tipos del filtro pueden ser combinadas. Se debe prestar atención

HTB (Hierarchical Token Bucket)

Es un algoritmo utilizado para dividir el ancho de banda según nos interese de manera que se puede especificar un *ancho de banda mínimo* y un *ancho de banda máximo*. De manera que el algoritmo asegura la disponibilidad del mínimo, y en caso de que haya ancho de banda libre se puede llegar hasta el máximo.

HTB tiene una funcionalidad muy similar a la de CBQ, aunque su implementación es completamente distinta y su configuración mucho más sencilla. El único 'pero' es que HTB aún no forma parte del kernel estándar de Linux y hay que parchearlo y recompilarlo para utilizarla.

SFQ (Stochastic Fairness Queueing)

Es una implementación sencilla de la familia de algoritmos de colas justas *(fair queueing)*.

En este algoritmo el tráfico se divide en un número bastante grande de colas FIFO, una por cada conversación. Entonces se envía el tráfico de una manera parecida a round robin, dando a cada sesión por turnos la oportunidad de enviar datos.

Es decir es un algoritmo que busca una equidad entre todas las peticiones para repartir de la manera más justa posible el ancho de banda.

TBF

La cubeta de fichas (Token Bucket Filter, TBF) funciona como un sistema de almacenamiento de crédito. Para que se pueda transmitir tráfico (octetos o paquetes) hay que retirar el correspondiente crédito de fichas (*tokens*) de una cubeta (*bucket*), si a la llegada del tráfico no hay fichas disponibles, éste es descartado (función policía).

Durante la operación de TBF puede darse una de las siguientes situaciones:

- Los datos llegan a TBF a una tasa que es igual a la de *tokens* entrantes.
  En este caso, cada dato entrante tiene su *token* correspondiente y pasa a la cola sin retrasos.
- Los datos llegan al TBF a una tasa menor a la de los *token*.
  Sólo una parte de los *tokens* se borran con la salida da cada dato que se envía fuera de la cola, de manera que se acumulan los *tokens*, hasta llenar la cubeta (*bucket*). Los *tokens* sin usar se pueden utilizar para enviar datos a velocidades mayores de la tasa de *tokens*, en cuyo caso se produce una corta ráfaga de datos.
- Los datos llegan al TBF a una tasa mayor a la de los *token*.

Esto significa que la cubeta se quedará pronto sin *tokens*. Si siguen llegando paquetes, empezarán a ser descartados. Esta última situación es muy importante, porque permite ajustar administrativamente al ancho de banda disponible a los datos que están pasando por el filtro.

La implementación de Linux añade una cola a la entrada, dando lugar a luna cierta capacidad de espera (y el consiguiente conformado de tráfico).

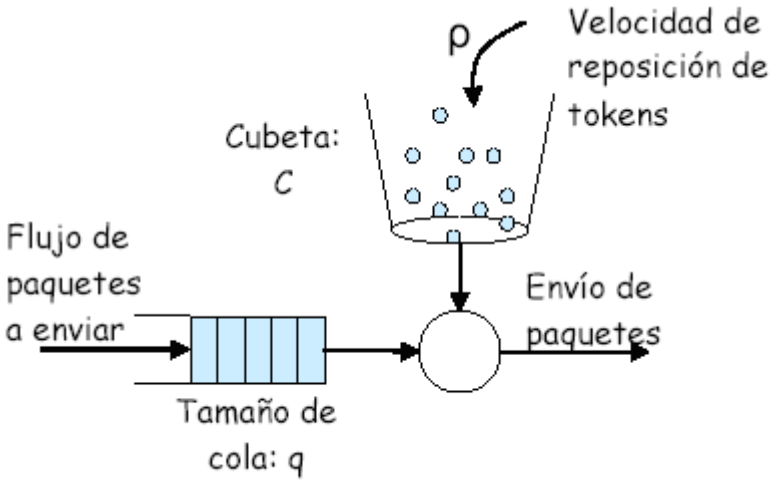

Figura 4 Modelo de Token Bucket

Los parámetros que definen el funcionamiento de la disciplina de cola TBF en Linux son los siguientes:

Tasa (*rate*) = ñ. Es la velocidad a la cuál se reponen los *tokens*. Representa la velocidad media de transmisión de un flujo dado.

Tamaño de la cubeta (*Burst / Buffer / Maxburst*) = C. Es el número de *tokens* que se pueden almacenar. En Linux un *token* es equivalente a un octeto de tráfico. Inicialmente la cubeta está llena de *tokens*. Como se estudió en teoría, si la tasa de generación de *tokens* es inferior a la velocidad del enlace, el tamaño de la cubeta es aproximadamente el tamaño de la ráfaga máxima.

Límite (*limit*). Es la suma de tamaño de la cubeta (*Burst / Buffer /Maxburst*) y del tamaño de la cola (*queue*), medido en octetos. Define implícitamente el tamaño de la cola de espera.

Si *limit* es igual *buffer*, no hay cola de espera y los paquetes que no se ajusten al perfil son descartados. Si es mayor, existirá una cola de espera a la entrada y, como resultado, se producirá conformado de tráfico.

Otra forma muy buena de controlar el ancho de banda es por medio del squid. A continuación se dará una explicación clara de cómo controlar ancho de banda con squid.

SQUID

Squid es probablemente el servidor proxy HTTP más avanzado disponible para Linux. Puede ayudarnos a ahorrar ancho de banda de dos maneras:

- La principal es una característica común a todos los servidores proxy guardan las páginas descargadas, las imágenes y otros objetos en memoria o en un disco. Por esto, si dos personas solicitan la misma página no se descargará de Internet sino del proxy local.
- Aparte del cacheado normal, Squid dispone de una prestación especial conocida como [pools: colas?] con retraso. Gracias a las [pools] con retraso es posible limitar el tráfico de internet de una manera razonable dependiendo de las llamadas 'palabras mágicas' existentes en cualquier URL.

Por ejemplo, una palabra mágica podría ser '.mp3', '.exe' o '.avi', etc. Cualquier parte distintiva de una URL (como .avi) puede definirse como una palabra mágica.

Con eso podemos decirle a Squid que descargue este tipo de archivos a una velocidad específica Si los usuarios de nuestra LAN descargan archivos al mismo tiempo lo harán entre todos, dejando así ancho de banda suficiente para las páginas web, el correo, las news, el irc, etc.

Instalar Squid con la prestación delay pools

Como se menciono, Squid tiene una prestación que se conoce como delay pools y que nos permite controlar el ancho de banda de bajada. Desgraciadamente, en la mayoría de las distribuciones Squid viene sin esa característica.
Así que si tiene Squid ya instalado tendrá que desinstalarlo e instalarlo de nuevo con las delay pools activadas de la manera que se explica más adelante.

1. Para obtener un mejor rendimiento de nuestro proxy Squid lo mejor es crear una partición aparte para su caché y llamarla /cache/. Su tamaño debería ser de alrededor de 300 megabytes dependiendo de sus necesidades.
Si no sabe cómo crear una partición puede crear el directorio /cache/ en su partición raíz pero el rendimiento de Squid podrá disminuir notablemente.

2. Añadimos un usuario 'squid' seguro:
useradd -d /cache/ -r -s /dev/null squid >/dev/null 2>&1
Nadie puede entrar en el sistema como squid, ni siquiera root.

3. Descargamos las fuentes de Squid de http://www.squid-cache.org

Cuando estaba escribiendo este COMO la última versión era Squid 2.4 estable:http://www.squid-cache.org/Versions/v2/2.4/squid-2.4.STABLE1-src.tar.gz

4. Desempaquetamos todo en /var/tmp:
5. # tar xzpf squid-2.4.STABLE1-src.tar.gz
6. Compilamos e instalamos Squid (todo va en una sola línea):

```
# ./configure --prefix=/opt/squid --exec-prefix=/opt/squid
enable-delay-pools --enable-cache-digests --enable-poll
disable-ident-lookups --enable-truncate --enable-removal-policies
# make all
# make install
```

Configurar Squid para poder usar la prestación de las delay pools

Configuramos nuestro archivo squid.conf (localizado en /opt/squid/etc/squid.conf):

```
#squid.conf
```

Todas las opciones de este archivo se encuentran muy bien documentadas en el propio squid.conf así como en
http://www.visolve.com/squidman/Configuration%20Guide.html
Los puertos por los que escuchará nuestro Squid.

```
http_port 8080
icp_port 3130
```

Los cgi-bin no se cachearán.

```
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
```

La memoria que usará Squid. Bueno, Squid usará mucha más que ésa.

```
cache_mem 16 MB
```

250 significa que Squid usará 250 megabytes de espacio en disco.

```
cache_dir ufs /cache 250 16 256
```

Lugares en los que irán los archivos de bitácora de Squid.

```
cache_log /var/log/squid/cache.log
cache_access_log /var/log/squid/access.log
cache_store_log /var/log/squid/store.log
cache_swap_log /var/log/squid/swap.log
```

Cuántas veces rotar los archivos de bitácora antes de borrarlos.
Acuda a la FAQ para más información.

```
logfile_rotate 10
redirect_rewrites_host_header off
cache_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 20 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
maximum_object_size 3000 KB
store_avg_object_size 50 KB
```

Configure esto si quiere que su proxy funcione de manera transparente. Eso significa que por lo general no tendrá que configurar todos los navegadores de sus clientes, aunque tiene algunos inconvenientes. Si deja esto sin comentar no pasará nada peligroso.

```
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Todos los usuarios de nuestra LAN serán vistos por los servidores web externos como si usasen Mozilla en Linux.

```
anonymize_headers deny User-Agent fake_user_agent Mozilla/5.0 (X11;
U; Linux i686; en-US; rv:0.9.6+) Gecko/20011122
```

Para acelerar aún más nuestra conexión ponemos dos líneas similares a las de más abajo. Apuntarán a un servidor proxy [parent] que usará nuestro propio Squid. No olvide cambiar el servidor por uno más rápido para usted.

Puede utilizar ping, traceroute y demás herramientas para comprobar la velocidad. Asegúrese de que los puerto http e icp son los correctos.

Descomente las líneas que comienzan por "cache_peer" de ser necesario.

Éste es el proxy que va a usar para todas las conexiones...

```
#cache_peer w3cache.icm.edu.pl parent 8080 3130 no-digest default
```

excepto para las direcciones e IPs que comiencen por "!". No es buena idea usar un mayor

cache_peer_domain w3cache.icm.edu.pl !.pl !7thguard.net !192.168.1.1

Esto resulta útil cuando queremos usar el Cache Manager.

Copie cachemgr.cgi al cgi-bin de su servidor web. Podrá acceder a él una vez lo haya hecho introduciendo en un navegador la dirección http://su-servidor-web/cgi-bin/cachemgr.cgi

cache_mgr your@email

cachemgr_passwd secret_password all

Éste es el nombre de usuario con el que trabajará nuestro Squid.

cache_effective_user squid

cache_effective_group squid

log_icp_queries off

buffered_logs on

DELAY POOLS

Ésta es la parte más importante para configurar el tráfico entrante con Squid. Para una descripción detallada acuda al archivo squid.conf o a la documentación de http://www.squid-cache.org.

No queremos limitar las descargas en nuestra red local.

acl magic_words1 url_regex -i 192.168

Queremos limitar la descarga de este tipo de archivos. Ponga todo esto en una única línea

acl magic_words2 url_regex -i ftp .exe .mp3 .vqf .tar.gz .gz .rpm .zip .rar .avi .mpeg ..ram .rm .iso .raw .wav .mov

No bloqueamos .html, .gif, .jpg y archivos similares porque por lo general no consumen demasiado ancho de banda.

Queremos limitar el ancho de banda durante el día permitiendo el ancho de banda completo durante la noche.

Cuidado! con el acl de abajo sus descargas se interrumpirán a las 23:59. Lea la FAQ si quiere envitarlo.

acl day time 09:00-23:59

Tenemos dos delay_pools diferentes. Acuda a la documentación de Squid para familiarizarse con delay_pools y delay_class.

delay_pools 2

Primer delay pool.

No queremos retrasar nuestro tráfico local. Hay tres cases de pools; aquí sólo hablaremos de la segunda.

Primera clase de retraso (1) de segundo tipo (2).

```
delay_class 1 2
#-1/-1 significa que no hay límites.
delay_parameters 1 -1/-1 -1/-1
#magic_words1: 192.168 que ya hemos puesto antes
delay_access 1 allow magic_words1
```

Segundo delay pool.

Queremos retrasar la descarga de los archivos mencionados en magic_words2.

Segunda clase de retraso (2) de segundo tipo (2).

```
delay_class 2 2
```

Los números siguientes son valores en bytes;

Debemos recordar que Squid no tiene en cuenta los bits de inicio/parada

```
5000/150000 son valores para la red al completo
5000/120000 son valores para la IP independiente
```

una vez los archivos descargados exceden los 150000 bytes, (o el doble o el triple) las descargas proseguirán a 5000 bytes/s

```
delay_parameters 2 5000/150000 5000/120000
```

Ya hemos configurado antes el día de 09:00 a 23:59.

```
delay_access 2 allow day
delay_access 2 deny !day
delay_access 2 allow magic_words2
#EOF
```

Cuando lo hayamos configurado todo debemos asegurarnos de que todo lo que se encuentre en los directorios /opt/squid y /cache pertenece al usuario 'squid'.

```
# mkdir /var/log/squid/
# chown squid:squid /var/log/squid/
# chmod 770 /var/log/squid/
# chown -R squid:squid /opt/squid/
# chown -R squid:squid /cache/
```

Ahora ya está todo listo para ejecutar Squid. Cuando lo hagamos por primera vez tendremos que crear sus directorios de caché:

 # /opt/squid/bin/squid -z

Ejecutamos Squid y comprobamos si todo funciona correctamente. Una buena herramienta para hacer eso es IPTraf; puede encontrarla en http://freshmeat.net. Asegúrese de haber configurado el proxy adecuado en sus navegadores (192.168.1.1, puerto 8080 en nuestro ejemplo):

 # /opt/squid/bin/squid

Si todo funciona añadimos la línea /opt/squid/bin/squid al final de nuestros scripts de inicio.

Normalmente puede ser /etc/rc.d/rc.local. Otras opciones útiles de Squid son:

 # /opt/squid/bin/squid -k reconfigure (reconfigures Squid si hicimos cualquier cambio en el archivo squid.conf)

 # /opt/squid/bin/squid -help :) no hace falta explicar qué hace

También puede copiar cachemgr.cgi al directorio cgi-bin de su servidor web para utilizar un útil gestor de caché.

TC

La orden *tc* (*traffic controler*) es un programa de usuario que se utiliza para configurar todos los elementos de control de tráfico, disciplinas de cola, filtros y clases, así como las relaciones entre ellos. Es un comando con una sintaxis compleja y, desafortunadamente, no hay una documentación completa de su funcionamiento, aunque sí hay varias referencias útiles del mismo.

Hay que tener especial cuidado al especificar las unidades los parámetros del comando *tc*. Los sufijos b y bit indican respectivamente octetos (*bytes*) y bits. Así, por ejemplo, si queremos una velocidad de 1Mbps, pondremos rate 1Mbit no rate 1Mb (que indicaría 1Mbyte). Para capacidad de almacenamiento K indica 1024 y M 1024x1024. En este caso, para evitar confusiones se recomienda no utilizar los prefijos mencionados y expresar los valores calculados directamente en octetos (*bytes*).

En cuanto a unidades de tiempo, se usan: *s*, *sec* y *secs* para segundos, *ms*, *msec* y *msecs* para milisegundos y *us*, *usec* y *usecs* para microsegundos.

Para el manejo de una disciplina de cola la sintaxis a emplear es la siguiente:

tc qdisc
[add|del|replace|change|get]
dev STRING
[handle QHANDLE]
[root|parent CLASSID]
[[QDISC_KIND] [help|OPTIONS]]

Las mayúsculas indican los valores que se le da al parámetro. Los parámetros significan:

- El primer parámetro sirve para crear, borrar, reemplazar, etc., la disciplina de cola, que recordemos que se denomina *qdisc*.
- *dev STRING* indica el interfaz sobre el que se crea la *qdisc*, por ejemplo *eth0*.
- *handle QHANDLE* es un manejador que puede ser usado después, en comandos de configuración para referirse a esta disciplina de cola.

El manejador de este *qdisc* tiene dos partes, número mayor y menor. Es habitual llamar a *qdisc* raíz como 1:0, el número menor de una *qdisc* es siempre 0. Las clases deben tener el mismo número mayor de la *qdisc* padre en la que se encuentran. No puede haber dos *qdisc* con el mismo andel. Las disciplinas de cola y clases se ordenan de forma jerárquica partiendo de una raíz común.

root
1:0
1:1
10:0 11:0 12:0
10:1 10:2 12:1 12:2

- *root* indica que la *qdisc* del dispositivo es la raíz del árbol jerárquico.
- *parent* representa el manejador de la *qdisc* padre, dentro de la cuál está *qdisc.*
- *QDISC_KIND* es el tipo de disciplina de cola como: *dsmark, tbf, cbq, red,* etc.
- *OPTIONS* son los parámetros de la disciplina de cola que son específicos de cada una de ellas.

Para visualizar la *qdisc* asociada a un dispositivo y sus estadísticas se usan las siguientes órdenes:

tc qdisc show [dev STRING]
tc qdisc –s [dev STRING]

2.10 SEGURIDAD

Seguridad en la Web

Dado el gran auge que hoy en día tiene Internet, su uso se ha masificado enormemente. Desde páginas meramente informativas hasta sitios interactivos usando tecnologías nuevas.

Empresas de diversa índole ya usan la Internet para comunicarse y el problema principal que surgió es la confiabilidad en que lo que se esta comunicando no sea visto por personas que puedan hacer mal uso de dicha información.

Por ejemplo, las tiendas comerciales ya están dando la posibilidad de realizar compras por la Web, pero el principal talón de Aquiles lo constituye la inseguridad que causa dar un número de tarjeta de crédito para pagar la compra.
O cosas tan simples como cuando uno envía un mail y no quiere que nadie lo lea sino el destinatario.
A raíz de todo esto surgieron tecnologías que persiguen mejorar la seguridad de todas estas comunicaciones.

Seguridad en la transmisión

La seguridad de este tipo se basa en el hecho de poder encriptar los mensajes que se envían por a red entre un servidor y un cliente y que solo ellos puedan descifrar los contenidos a partir de una clave común conocida solo por los dos.

2.10.1 PROTOCOLOS

Para llevar a cabo esta seguridad se crearon diversos protocolos basados en esta idea:

- SSH: Usado exclusivamente en reemplazo de telnet
- SSL: Usado principalmente en comunicaciones de hipertexto pero con posibilidad de uso en otros protocolos
- TSL: Es del mismo estilo del anterior.
- HTTPS: Usado exclusivamente para comunicaciones de hipertexto
- IPSEC: Protege todos los paquetes de IP en la capa de red, forma una capa segura de un nodo de la red a otro.
- IKE: se usa para establecer una asociación de seguridad (Security Association- SA) entre dos pares.

SSH (Secure Shell)

Este protocolo fue diseñado para dar seguridad al acceso a computadores en forma remota.
Cumple la misma función que telnet o rlogin pero además, usando criptografía, logra seguridad con los datos.
A diferencia de telnet u otro servicio similar, SSH utiliza el puerto 22 para la comunicación y la forma de efectuar su trabajo es muy similar al efectuado por SSL.
Para su uso se requiere que por parte del servidor exista un demonio que mantenga continuamente en el puerto 22 el servicio de comunicación segura, el sshd.

El cliente debe ser un software tipo TeraTerm o Putty que permita la hacer pedidos a este puerto 22 de forma cifrada.
La forma en que se entabla una comunicación es en base la misma para todos los protocolos seguros:

- El cliente envía una señal al servidor pidiéndole comunicación por el puerto 22.
- El servidor acepta la comunicación en el caso de poder mantenerla bajo encriptación mediante un algoritmo definido y le envía la llave pública al cliente para que pueda descifrar los mensajes.
- El cliente recibe la llave teniendo la posibilidad de guardar la llave para futuras comunicaciones o destruirla después de la sesión actual.
- Se recomienda que si se esta en un computador propio, la clave sea guardada, en otro caso, destruirla

SSL (Secure Socket Layer) y TLS(Transport Layer Secure)

El protocolo SSL fue desarrollado por Netscape para permitir confidencialidad y autenticación en Internet. SSL es una capa por debajo de HTTP y tal como lo indica su nombre esta a nivel de socket por lo que permite ser usado no tan solo para proteger documentos de hipertexto sino también servicios como FTP, SMTP, TELNET entre otros.

La idea que persigue SSL es encriptar la comunicación entre servidor y cliente mediante el uso de llaves y algoritmos de encriptación.

El protocolo TLS esta basado en SSL y son similares en el modo de operar. Es importante señalar que ambos protocolos se ejecutan sobre una capa de transporte definida, pero no determinada. Esto indica que pueden ser utilizados para cualquier tipo de comunicaciones. La capa de transporte más usada es TCP cobre la cual pueden implementar seguridad en HTTP.

Como punto de diferencia se puede mencionar que existen protocolos implementados sobre la capa de red, por ejemplo sobre IP. Tal es el caso de IPSec.

Estructura de los protocolos

Estos protocolos se componen de dos capas: el Record Protocol y el Handshake Protocol.

El Record Protocol es la capa inmediatamente superior a TCP y proporciona una comunicación segura. Principalmente esta capa toma los mensajes y los codifica con algoritmos de encriptación de llave simétrica como DES, RC4 aplicándole una MAC (Message Authentication Code) para verificar la integridad, logrando así encapsular la seguridad para niveles superiores.

El Handshake protocol es la capa superior a la anterior y es usada para gestionar la conexión inicial.

Funcionamiento

En resumidas cuentas, después que se solicita una comunicación segura, servidor y el cliente se deben poner de acuerdo en como se comunicaran (SSL Handshake) para luego comenzar la comunicación encriptada. Luego de terminada la transacción, SSL termina.

Solicitud de SSL:

Típicamente este proceso ocurre en el momento que un cliente accede a un servidor seguro, identificado con "https://...". pero como se mencionó, no necesariamente es usado para HTTP. La comunicación se establecerá por un puerto distinto al utilizado por el servicio normalmente. Luego de esta petición, se procede al SSL Handshake.

SSL Handshake:

En este momento, servidor y cliente se ponen de acuerdo en varios parámetros de la comunicación. Se puede dividir el proceso en distintos pasos:

- Client Hello: El cliente se presenta. Le pide al servidor que se presente (certifique quien es) y le comunica que algoritmos de encriptación soporta y le envía un número aleatorio para el caso que el servidor no pueda certificar su validez y que aun así se pueda realizar la comunicación segura.
- Server Hello: El servidor se presenta. Le responde al cliente con su identificador digital encriptado, su llave pública, el algoritmo que se usará, y otro número aleatorio. El algoritmo usado será el más poderoso que soporte tanto el servidor como el cliente.
- Aceptación del cliente: El cliente recibe el identificador digital del servidor, lo desencripta usando la llave pública también recibida y

verifica que dicha identificación proviene de una empresa certificadora segura. Luego se procede a realizar verificaciones del certificado (identificador) por medio de fechas, URL del servidor, etc. Finalmente el cliente genera una llave aleatoria usando la llave pública del servidor y el algoritmo seleccionado y se la envía al servidor.

- Verificación: Ahora tanto el cliente y el servidor conocen la llave aleatoria (El cliente la generó y el servidor la recibió y desencriptó con su llave privada). Para asegurar que nada ha cambiado, ambas partes se envían las llaves. Si coinciden, el Handshake concluye y comienza la transacción.

Intercambio de Datos:

Desde este momento los mensajes son encriptados con la llave conocida por el servidor y el cliente y luego son enviados para que en el otro extremo sean desencriptados y leídos.

Terminación de SSL

Cuando el cliente abandona el servidor, se le informa que terminara la sesión segura para luego terminar con SSL.

En el siguiente esquema se muestra todo el proceso del Handshake:

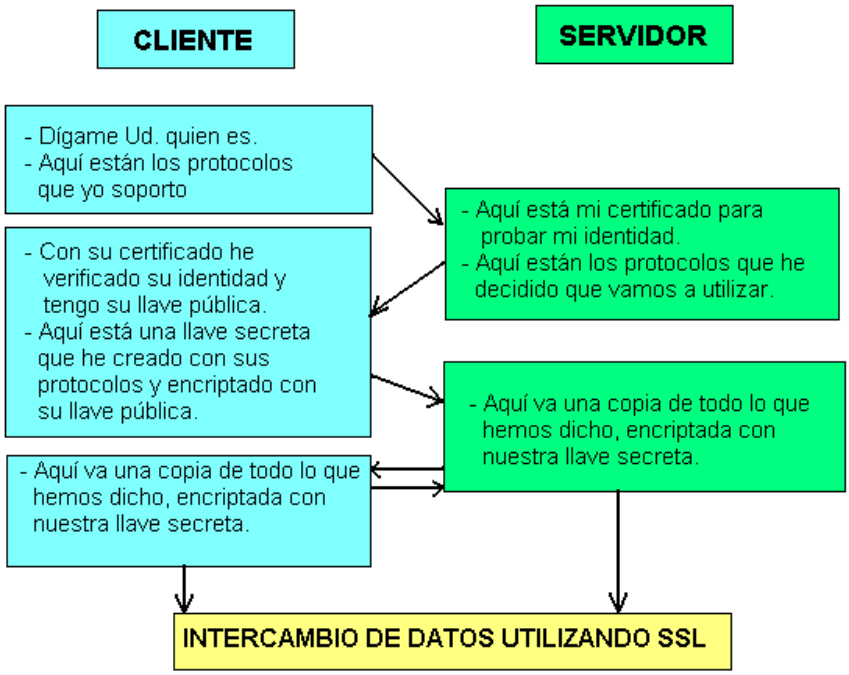

Figura 5 Esquema del Handshake

Protocolo HTTPS

Uno de los usos comunes de ssl es el de establecer una comunicación Web segura entre un browser y un Webserver. Es aquí donde se usa https que es básicamente http sobre ssl con un esquema de invocación por medio de url. Es importante hacer notar que el uso del protocolo https no impide en caso alguno que se pueda utilizar http, por lo que la mayoría de los browsers advierten cuando una página tiene elementos que no son seguros en entornos seguros, como también advierten cuando se invoca un protocolo distinto al de la pagina actual .

Protocolo IPSec

El protocolo de seguridad de IP (IPsec), que es una tecnología que protege todos los paquetes de IP en la capa de red, forma una capa segura de un nodo de la red a otro. IPsec puede incluso ser usado para crear VPN basados en IP.

Sin embargo, el protocolo no estipula cómo se han de autentificar los pares ni cómo se intercambian las claves de la sesión. Estas tareas son gestionadas por el protocolo de intercambio de claves de Internet (Internet key exchange - IKE).

Para cifrar todos los paquetes, incluyendo los paquetes sin conexión y los paquetes de control de IP, IPsec es una buena elección. IPsec es también preferible para conexiones generales de capa de redes seguras, tales como VPN.

IPsec surge como un estándar de seguridad que añade estas dos funcionalidades a los protocolos de red, opcional en IPv4 y obligatorio en las implementaciones Ipv6. Este estándar ha sido implementado por distintos productos permitiendo la interoperabilidad entre ellos, es muy conocida por ejemplo la implementación de CISCO, aunque otras compañías conocidas como Lucent o 3COM han hecho su propia implementación de IPsec.

El funcionamiento de IPsec se basa en túneles encriptados y autenticados. Por un lado se establece un túnel previa autenticación con clave 30 publica entre dos maquinas y posteriormente se envían los datos a través del túnel de modo que dentro del segmento de datos del paquete \normal" de red, incluimos un paquete de datos completo encriptado en origen y que será desencriptado en destino (y rutado si es necesario).

Mediante este mecanismo convertimos una red potencialmente insegura como internet, en una red segura sobre la que se pueden implantar ciertos servicios que hasta este momento serian impensables sobre una red de este tipo.

Aplicaciones de IPsec

Como hemos dicho, IPsec abre un abanico sumamente interesante de posibilidades, las dos principales son las Redes Privadas Virtuales (VPN) y los
\RoadWarriors".

Las Redes Privadas Virtuales, permiten comunicar varias redes locales entre si mediante el uso de un medio inseguro como es internet usando sus direcciones IP locales como si estuviesen dentro de nuestra red físicamente y
siempre de un modo seguro.

Como \RoadWarriors" se denominan a aquellos puntos itinerantes de nuestra red (comerciales con portátiles, trabajadores que realizan su trabajo desde casa, conexiones desde hoteles, etc.) a los que le aseguramos que la comunicación entre el punto en el que se encuentre y su lugar de trabajo, a través de internet, se realizara de un modo seguro sin ser posible que sea interceptada provocando el descubrimiento de datos sensibles (generalmente de trabajo).

Protocolo IKE

El protocolo de Intercambio de clave de Internet (Internet key exchange - IKE) se usa para establecer una asociación de seguridad (Security Association- SA) entre dos pares. Un SA es un secreto compartido (junto con una normativa para el secreto) entre las partes comunicantes. El SA se necesita para proteger la comunicación real entre pares. IKE se usa por lo general para negociar un SA para IPsec. IKE se basa en el protocolo de asociación para la seguridad y gestión de claves de Internet (Internet security association and key management protocol - ISAKMP), que sugiere una negociación de clave basada en dos fases diferentes:

Fase I

Los dos pares establecen un canal seguro para la ulterior comunicación negociando los SA de ISAKMP.

Fase II

Bajo la protección del SA negociado en la Fase I, los pares negocian los SA que pueden ser usados para proteger la comunicación real; o sea, el SA de IPsec.

IKE define dos modos en la Fase I:

- El MODO PRINCIPAL (MAIN MODE) da intercambio de clave autentificada con protección de identidad.
- El MODO AGRESIVO (AGRESSIVE MODE) da intercambio de clave autentificada sin protección de identidad.

Para la Fase I, IKE define (para los modos principal y agresivo) cuatro métodos de autentificación diferentes:

1. autentificación con firmas;
2. autentificación con cifrado de clave pública;
3. autentificación con un modo revisado de cifrado de clave pública; y
4. autentificación con una clave previamente compartida.

En los métodos 2, 3 y 4, se supone que el iniciador de la negociación de clave ya ha recibido del respondedor la clave pública o una clave previamente compartida. La Figura muestra la autentificación IKE con el protocolo de firma para los modos principal y agresivo. Los diferentes campos del protocolo son como sigue:

- HDR—el campo de cabecera incluye un cookie aleatorio elegido por el iniciador y el respondedor.
- SA - el campo de asociación de seguridad incluye varios parámetros junto con una propuesta para los atributos criptográficos que el par quiere usar durante las negociaciones IKE. El iniciador envía las propuestas; el respondedor elige entre éstas y devuelve una nueva SA.
- KE—el valor del intercambio de clave pública.
- N—un valor aleatorio usado para calcular materiales de clave compartidos por los pares.
- ID—un campo de identidad; por ejemplo, una dirección Ipv4.
- CERT—un certificado que contiene una clave de comprobación de firma.
- SIG—Una firma digital calculada sobre un valor arbitrario (hash). El valor hash del iniciador se obtiene de los valores del cookie del iniciador y del respondedor, de un “secreto premaster,” de los valores KE, del valor SA, y de la ID del iniciador. El valor hash del respondedor se obtiene

exactamente de la misma manera pero usando el valor de ID del respondedor.

IKE Fase II tiene solamente un modo obligatorio:
El MODO RÁPIDO (QUICK MODE). IKE Fase II es usa exclusivamente para negociar los parámetros de seguridad para otro protocolo tal como IPsec.
No hay certificados involucrados en esta fase.

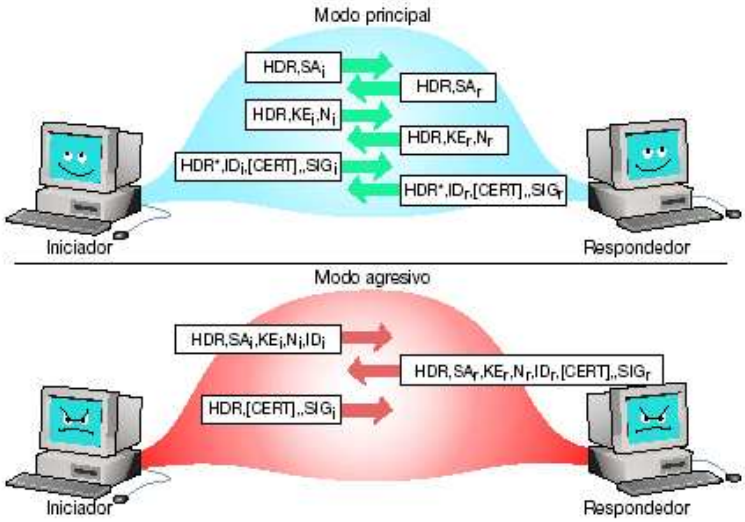

Figura 6 IKE fase II

2.10.2 ENCRIPTACIÓN

Encriptación es el proceso mediante el cual cierta información o "texto plano" es cifrado de forma que el resultado sea ilegible a menos que se conozcan los datos necesarios para su interpretación. Es una medida de seguridad utilizada para que al momento de almacenar y/o transmitir información sensible ésta no pueda ser obtenida con facilidad por terceros. Opcionalmente puede existir además un proceso de desencriptación a través del cuál la información puede ser interpretada de nuevo a su estado original, aunque existen métodos de encriptación que no pueden ser revertidos.

Algunos de los usos más comunes de la encriptación son el almacenamiento y transmisión de información sensible como contraseñas, números de identificación legal, números de tarjetas de crédito, reportes administrativo-contables y conversaciones privadas, entre otros.

La encriptación hace uso de diversas fórmulas matemáticas con el propósito de transformar el texto plano en un criptograma el cual es un conjunto de caracteres que

a simple vista no tiene ningún sentido para el lector. La mayoría de los métodos de encriptación utilizan una clave como parámetro variable en las mencionadas fórmulas matemáticas de forma que a pesar de que un intruso las conozca, no le sea posible descifrar el criptograma si no conoce la clave, la cual solo se encuentra en posesión de las personas que pueden tener acceso a la información en cuestión. Algunos métodos utilizan incluso dos claves, una privada que se utiliza para la encriptación y otra pública para la desencriptación. En algunos métodos la clave pública no puede efectuar la desencriptación o descifrado, sino solamente comprobar que el criptograma fue encriptado o cifrado usando la clave privada correspondiente y no ha sido alterado o modificado desde entonces.

La encriptación como proceso forma parte de la criptología, ciencia que estudia los sistemas utilizados para ocultar la información.

Aunque la criptología surgió con gran anterioridad, la informática ha revolucionado los métodos que se utilizan para la encriptación/desencriptación de información, debido a la velocidad con que las computadoras pueden realizar las fórmulas matemáticas requeridas para llevar a cabo estos métodos y a la complejidad que han alcanzado debido a este hecho.

Encriptación de 40-bits y 128-bits.

Existen varios niveles de encriptación, pero las combinaciones más comunes son 40-512 bits ("llave secreta--llave pública") y 128-1024 bits ("llave secreta--llave pública"). La versión 128-1024 bits es el tipo de encriptación más fuerte que existe en el mercado. Actualmente U.S.A prohíbe la exportación de productos con este tipo de Tecnología, pero cabe mencionar que ya existen varios productos producidos en Europa con esta Tecnología que no poseen tales restricciones de exportación.

La gran mayoría de los sitios en Internet utilizan la encriptación 40-512 bits, la encriptación 128-1024 bits es utilizada generalmente en transacciones de alto riesgo, como las bancarias.

2.10.3 CRIPTOGRAFIA

Entendemos por Criptografía (Kriptos=ocultar, Graphos=escritura) la técnica de transformar un mensaje inteligible, denominado texto en claro, en otro que sólo puedan entender las personas autorizadas a ello, que llamaremos criptograma o texto

cifrado. El método o sistema empleado para encriptar el texto en claro se denomina algoritmo de encriptación**.**

La Criptografía es una rama de las Matemáticas, que se complementa con el Criptoanálisis, que es la técnica de descifrar textos cifrados sin tener autorización para ellos, es decir, realizar una especie de Criptografía inversa. Ambas técnicas forman la ciencia llamada Criptología.

La base de las Criptografía suele ser la aplicación de problemas matemáticos de difícil solución a aplicaciones específicas, denominándose criptosistema o sistema de cifrado a los fundamentos y procedimientos de operación involucrados en dicha aplicación.

Criptografía simétrica

Incluye los sistemas clásicos, y se caracteriza por que en ellos *se usa la misma clave* para encriptar y para desencriptar, motivo por el que se denomina simétrica.

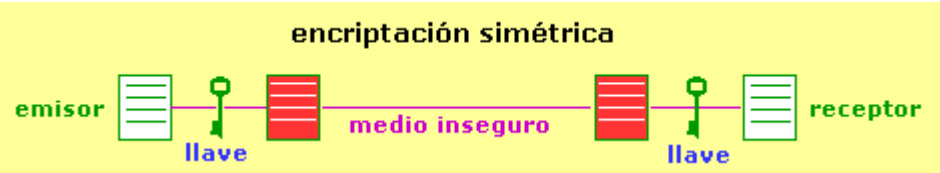

Figura 7 Encriptación Geométrica

Toda la seguridad de este sistema está basada en la llave simétrica, por lo que es misión fundamental tanto del emisor como del receptor conocer esta clave y mantenerla en secreto. Si la llave cae en manos de terceros, el sistema deja de ser seguro, por lo que habrá que desechar dicha llave y generar una nueva.

Para que un algoritmo de este tipo sea considerado fiable debe cumplir varios requisitos básicos:

1. conocido el criptograma (texto cifrado) no se pueden obtener de èl ni el texto en claro ni la clave.
2. conocidos el texto en claro y el texto cifrado debe resultar más caro en tiempo o dinero descifrar la clave que el valor posible de la información obtenida por terceros.

Generalmente el algoritmo de encriptación es conocido, se divulga públicamente, por lo que la fortaleza del mismo dependerá de su complejidad interna y sobre todo de la longitud de la clave empleada, ya que una de las formas de criptoanálisis primario de

cualquier tipo de sistema es la de prueba-ensayo, mediante la que se van probando diferentes claves hasta encontrar la correcta.

Los algoritmos simétricos encriptan bloques de texto del documento original, y son más sencillos que los sistemas de clave pública, por lo que sus procesos de encriptación y desencriptacion son más rápidos.

Todos los sistemas criptográficos clásicos se pueden considerar simétricos, y los principales algoritmos simétricos actuales son **DES**, **IDEA** y **RC5**. Actualmente se están llevando a cabo un proceso de selección para establecer un sistema simétrico estándar, que se llamara **AES** (Advanced Encryption Standart), que se quiere que sea el nuevo sistema que se adopte a nivel mundial.

Las principales desventajas de los métodos simétricos son la distribución de las claves, el peligro de que muchas personas deban conocer una misma clave y la dificultad de almacenar y proteger muchas claves diferentes.

Criptografía de clave publica

También llamada asimétrica, se basa en el uso de dos claves diferentes, claves que poseen una propiedad fundamental: una clave puede desencriptar lo que la otra ha encriptado.
Generalmente una de las claves de la pareja, denominada clave privada, es usada por el propietario para encriptar los mensajes, mientras que la otra, llamada clave publica, es usada para desencriptar el mensaje cifrado.

Las claves publica y privada tienen características matemáticas especiales, de tal forma que se generan siempre a la vez, por parejas, estando cada una de ellas ligada intrínsecamente a la otra, de tal forma que si dos llaves publicas son diferentes, entonces sus llaves privadas asociadas también lo son, y viceversa.

Los algoritmos asimétricos están basados en funciones matemáticas fáciles de resolver en un sentido, pero muy complicadas de realizar en sentido inverso, salvo que se conozca la clave privada, como la potencia y el logaritmo. Ambas claves, publica y privada, están relacionadas matemáticamente, pero esta relación debe ser lo suficientemente compleja como para que resulte muy difícil obtener una a partir de la

otra. Este es el motivo por el que normalmente estas claves no las elige el usuario, si no que lo hace un algoritmo especifico para ello, y suelen ser de gran longitud.

Mientras que la clave privada debe mantenerla en secreto su propietario, ya que es la base de la seguridad del sistema, la clave pública es difundida ampliamente por Internet, para que este al alcance del mayor número posible de personas, existiendo servidores que guardan, administran y difunden dichas claves.

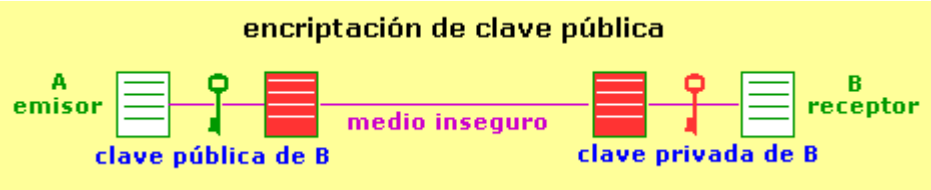

Figura 8 Encriptación Clave Pública

En este sistema, para enviar un documento con seguridad, el emisor (A) encripta el mismo con la clave publica del receptor (B) y lo envía por el medio inseguro. Este documento esta totalmente protegido en su viaje, ya que solo se puede desencriptar con la clave privada correspondiente, conocida solamente por B. Al llegar el mensaje cifrado a su destino, el receptor usa su clave privada para obtener el mensaje en claro.

2.10.4 FIRMA DIGITAL

Una firma digital, la cual no debe ser confundida con un certificado digital, es una firma electrónica que puede ser utilizada para autenticar la identidad del emisor de un mensaje, el firmante del documento o el dueño de una tarjeta de crédito. También puede ser utilizada para asegurar que el contenido original de un mensaje o documento enviado no ha sido cambiado. Una firma digital es usualmente generada desde un certificado digital usando una tecnología de llave pública y privada. (Es importante notar que una firma digital no sólo es una imagen grabada de una firma, un error relativamente común). Los beneficios adicionales de una firma digital son que es fácil de transportar, no puede ser fácilmente repudiada, no puede ser imitada por otra persona y puede ser automáticamente sellada a tiempo. Una firma digital puede ser utilizada con cualquier tipo de mensaje, ya sea codificado o no, de forma que el receptor puede estar seguro de la identidad del emisor del mensaje así como que el mensaje llegó intacto. Un certificado digital también contiene la firma digital de la autoridad emisora del certificado de forma que cualquiera puede verificar que el cerificado es real.

La firma convencional es usada cuando la comunicación es personal, si esta comunicación fuese por ejemplo por teléfono no es posible usar la firma convencional. La firma digital está precisamente diseñada para poder ser usada a grandes distancias, y principalmente cuando esta comunicación esta hecha por dos computadoras e Internet, además puede ser usada por muchos dispositivos electrónicos.

Cabe también mencionar, que aunque la firma convencional puede ser enviada vía fax o por un documento que copie el garabato, ésta no es válida legalmente. Esta firma convencional se usa solo por conveniencia de alguna corporación o institución, por ejemplo al usar un sello que estampa la firma de algún ejecutivo, es usada sólo por la rapidez que representa usarla, pero legalmente no es válida. Sólo es válida aquella que es derivada del puño y letra de la persona. Por su parte la firma digital garantiza ser mejor que la convencional y sería de gran beneficio si esta tuviese validez legal.

Quizá la mayor diferencia entre la firma convencional y la firma digital es que la primera en su método de verificación existe una gran probabilidad de error, según algunos hasta del 20%, y en el caso de la firma digital, este error es inapreciable. Es Es prudente mencionar que tipos de firma hay:

1) El método más usado para firmar digitalmente es el conocido como RSA, lo importante de este método es que es el más usado actualmente y por lo tanto es conveniente usarlo para poder ser compatible. Para que sea segura la longitud de sus claves (una pública y otra privada) debe de ser de 1024 bits, es decir un número de un poco más de 300 dígitos.

2) Otro método reconocido para firma digital es el llamado DSA, que es oficialmente aceptado para las transacciones oficiales en el gobierno de USA. Este método usa también claves del mismo tamaño que RSA, pero esta basado en otra técnica. Aún así, sea podido mostrar que es casi equivalente en seguridad a RSA.

3) Una tercera opción es el método que usa curvas elípticas, este método tiene la ventaja a los dos anteriores a reducir hasta en 164 bits, es decir como 45 dígitos las claves, manteniendo la misma seguridad. Por lo que es más propio para ser usado donde existen recursos reducidos como en Smart Cards, PDAs,

etc. Actualmente este método se ha integrado como el reemplazo oficial de DSA para el gobierno de USA.

4) Entre los posibles ataques a los anteriores métodos esta la posible remota construcción de una computadora cuántica, esta podría efectuar una cantidad tan grande de cálculos al mismo tiempo que podría romper los sistemas anteriores, incluso ya existen estos algoritmos que romperían los sistemas. Sin embargo ya existe otro método de forma que aún con la computación cuántica no existe aún algoritmo que pueda romperlos. Este sistema es que esta basado en lattices (retículas), se conoce como NTRU (Number Theory Research Unit) y entre otras cualidades es más eficiente que RSA.

2.10.5 CERTIFICADOS DIGITALES

Es un archivo de aproximadamente 1k de tamaño, que contiene, primero los datos del propietario, después su clave pública y la firma digital de una autoridad competente. Cuando una persona solicita un certificado digital, se generan su par de claves, la pública y la privada. La clave pública viene en el certificado digital explícitamente. La clave privada queda en custodia del propietario del certificado. El tercer elemento importante que tiene el certificado digital es la firma digital de una autoridad certificadora, quien esta como aval de que los datos corresponden al propietario. El certificado digital queda muy parecido entonces a un documento oficial de identificación como un pasaporte o una licencia de conducir. Otra importante característica del certificado digital es que contiene además de lo ya mencionado, El nombre de los algoritmos que se usan para la firma digital.

En la actualidad tenemos un formato (estándar) que se ha extendido casi para todas las aplicaciones, este es el llamado X.509. Este formato contiene los datos del poseedor del certificado, la clave pública del propietario, y la firma de una autoridad certificadora. La mejor propiedad del formato X.509 es que contiene el mínimo necesario de información para poder realizar muchas transacciones, principalmente comerciales y financieras. Sin embargo para otras aplicaciones puede ser un poco robusto.

1. **Certificados de Representación de Empresa/Entidad:** Expedición de certificados digitales en donde se relaciona una clave pública con una persona, indicando que una persona determinada se ha identificado como representante

legal de una persona jurídica, una Entidad del Estado, o como comerciante persona natural en el ámbito de su actividad profesional o mercantil. La clave privada correspondiente a la clave pública estará bajo la custodia permanente de la persona natural que figura en el certificado digital.

2. **Certificados de Pertenencia a Empresa/Entidad:** Expedición de certificados digitales en donde se relaciona una clave pública con una persona natural, indicando que una determinada persona se ha identificado como perteneciente a una determinada empresa o entidad del Estado, o como una persona que desempeña una determinada función o tiene una determinada relación respecto de la misma. La clave privada correspondiente a la clave pública estará bajo la custodia permanente de la persona natural que figura en el certificado digital.
3. **Certificados de Servidor Seguro:** Expedición de certificados digitales en los que se relaciona una clave pública con una dirección URL, indicando que una persona ha afirmado que un dominio en Internet está bajo el control de una cierta persona. TÉCNICAMENTE LOS CERTIFICADOS DE SERVIDOR SEGURO TIENEN LOS MISMOS USOS QUE LOS CERTIFICADOS DIGITALES DE REPRESENTACIÓN DE EMPRESA Y DE PERTENENCIA A EMPRESA. NO OBSTANTE, DESDE EL PUNTO DE VISTA JURÍDICO NO RESPALDAN FIRMAS DIGITALES.
4. **Certificados de Profesional Titulado:** Expedición de certificados digitales en donde se relaciona una clave pública con una persona natural, indicando que una determinada persona natural ha 1. afirmado: haber obtenido un título profesional debidamente reconocido en su República o en un Estado Extranjero, y 2. que ha obtenido el correspondiente registro, licencia, colegiatura o tarjeta profesional requerido para el ejercicio de su profesión en su República o en un Estado Extranjero. La clave privada correspondiente a la clave pública estará bajo la custodia permanente de la persona natural que figura en el certificado digital.
5. **Certificados para Firma de Código:** Expedición de certificados digitales en donde se relaciona una clave pública con una persona natural, indicando que una determinada persona natural o jurídica ha afirmado (i) adelantar las actividades de desarrollo, diseño, programación, mantenimiento, distribución de software, aplicativos, código fuente o código objeto, o afines y, (ii) tener el propósito de ser el originador de mensajes de datos que contengan información, software, aplicativos, código fuente o código objeto. La clave privada

correspondiente a la clave pública estará bajo la responsabilidad del representante legal de la persona jurídica.

6. **Certificados de VPN / Intranet:** Expedición de certificados digitales en donde se relaciona una clave pública con una persona natural, indicando que una persona natural o jurídica o entidad del Estado ha afirmado que, dentro de una red privada, tiene el control de una determinada máquina en dicha red privada. TÉCNICAMENTE LOS CERTIFICADOS DE VPN/INTRANET NO RESPALDAN JURIDICAMENTE, NI PUEDEN SER UTILIZADOS, PARA LA GENERACIÓN FIRMAS DIGITALES.
7. **Certificados de Titular de función Pública:** Expedición de certificados digitales en donde se relaciona una clave pública con una persona natural, indicando que una determinada persona natural ha afirmado: 1.Que ha sido nombrado o ser titular legal del cargo de notario, cónsul, juez de la república, magistrado o registrador en su República. 2. Que se encuentra en ejercicio del mismo. La clave privada correspondiente a la clave pública estará bajo la custodia permanente de la persona natural que figura en el certificado digital.
8. **Certificados de Persona Natural:** Expedición de certificados digitales en donde se relaciona una clave pública con una persona natural, indicando que una persona determinada se ha identificado plenamente con documento(s) de identidad válido(s) y vigente(s) expedidos por la autoridad competente de su República, o con documento(s) equivalente(s) expedido(s) por la autoridad competente de cualquier Estado Extranjero. La clave privada correspondiente a la clave pública estará bajo la custodia permanente de la persona natural que figura en el certificado digital.

CAPITULO III

ANALISIS DE SERVIDORES WEB (Apache e IIS)

3.1. ESTUDIO INTERNO DE LOS SERVIDORES WEB (APACHE E IIS)

3.1.1. APACHE

Apache estuvo en el momento adecuado y en el lugar adecuado. Los creadores de sitios web necesitaban ciertas opciones y reparaciones de fallos, por lo que nació Apache (software de los usuarios para los usuarios). El modelo Open Source era ideal para este proyecto, ya que, especialmente en los primeros días de la web, todo iba muy de prisa, y las empresas no se podían permitir el lujo de esperar a que un director de ingeniería decidiera si se podía vende un producto o un servicio por la red. Necesitaban tener una posibilidad inmediatamente, y tenían que hacer algo ellas mismas si no querían verse relegadas.

Apache se ejecuta en más sitios web que si sumamos el resto de servidores, ya que es un software mejor. Algunas personas prefieren Apache a otros servidores por que es gratuito. Pero, incluso en organizaciones donde el precio es secundario, como IBM, Apache es el elegido.

Hay un viejo dicho en la industria del software: "Bueno, rápido y barato: elija cualquiera da los dos". El Proyecto Apache, en cierto modo, ha aunado las tres características.

Al Comienzo.

La Web sigue siendo un fenómeno muy reciente. Tim Berners-Lee inventó la Web a finales de los noventas cuando trabajaba en el CERN, el Laboratorio Europeo de Física de Partículas. Lo desarrollo para que los físicos de varias universidades de todo el mundo tuvieran un acceso instantáneo a la información, para permitirles colaborar en una serie de proyectos.

Tim definió los URL, HTTP y HTML y, con Robert Cailliau, escribió su primer servidor web y el primer software para clientes web, que posteriormente se bautizó con el nombre de navegador.

Poco después del trabajo inicial de Tim, un grupo de Centro Nacional de Actividades de Supercomputación (NCSA), de la Universidad de Illinois, en Urbana-Champaign (UIUC) desarrolló el servidor web HTTPd NCSA y el navegador web gráfico NCSA Mosaic. Mosaic no fue el primer navegador web gráfico, aunque lo crea. Ese honor le corresponda a Viola, escrito por PeiWei y disponible con anterioridad a Mosaic se apropió rápidamente de esta condición, y se convirtió en el navegador web más usado en 1992.

Antes de que el código fuente del servidor se pusiera a disposición de todo el mundo, muchos de sus usuarios habituales habían desarrollado sus propias soluciones a los errores y sus propias características. Estas soluciones se com partían fortuitamente a través de Usenet, pero no había un mecanismo central que recuperara y distribuyera dichas soluciones.

Por consiguiente, Apache (al igual que la World Wide Web) fue ensamblado por voluntarios, aunque la terminación del proyecto HTTPd NCSA dejará a los desarrolladores con un producto que no funciona bien en la época y con nadie a quien reclamar, al final se consiguió un producto muy superior.

***Responsabilidad*.**

En febrero de 1995, Brian Behlendorf y Cliff Skolnick ensamblaron una lista de envío, prepararon una computadora y consiguieron ancho de banda, donado por HotWired. Brian construyó un árbol CVS (Sistema de Versiones Simultáneas), en virtud del cual todo el que quisiera podía contribuir a crear nuevas características y a reparar errores. De esta forma, un grupo de desarrolladores podían recoger las modificaciones a sus códigos y crear un producto combinado, comenzando con HTTPd 1.3 NCSA, empezaron a aplicar estas soluciones. La primera versión de este producto, llamado Apache, fue la versión 0.6.2, lanzada en abril de 1995.

Poco después del primer lanzamiento, Thau diseñó una arquitectura completamente nueva. Comenzando con la versión 0.8.8 en agosto de 1995, Apache se incorporó a esta nueva base de código.

Netcraft muestra que Apache sobrepasa a NCSA como primer servidor http a principios de 1996.

Funcionamiento

Apache es un producto fantástico. Hace todo lo que se quiere que haga, y nada de lo que no se quiere. Es rápido, fiable y barato. ¿Qué más se podría pedir de una unidad de software?

Apache puede ser todo esto porque es open source. Esto quiere decir que todo el que utilice el producto tiene acceso al código fuente. Si tiene una idea de algo que podría ser útil, puede escribir y entregar el código al grupo apache para su posible inclusión en el producto. Esto significa que las peticiones de apache son las prestaciones que la gente de a pie están utilizando en sus sitios web, y no las prestaciones que alguien ha sugerido en una reunión de marketing.

Además, cuando se encuentran fallos, las numerosas personas que tienen acceso al código podrán sugerir soluciones al problema (o por, citar a Eric Raymond, "con tantos ojos, cualquier fallo es superficial"). De ahí que la reparación de fallos siga rápidamente a su aparición. Esto contrasta con, los productos de software cerrados, donde, si se informa de un fallo, se está a merced del programa del responsable para esa reparación de fallos (en el caso de que atendieran su observación).

3.1.2 REQUERIMIENTOS DEL SISTEMA

Los requerimientos del sistema para ejecutar Apache son muy pocos, Necesitará un mínimo de 12MB de espacio temporal en la unidad de disco duro para el proceso de instalación. Tras la instalación, Apache ocupa cerca de 3MB, además el espacio que se utilice para colocar el contenido web.

También necesita un compilador ANSI-C. el compilador GNU C, que se conoce como GCC, es el compilador reconocido, pero otros compiladores también trabajan bien si son compilables con ANSI-C.

Perl (versión 5.003 o superior) es necesario en algunos de los scripts de soporte opcionales. El soporte para Objetos Compartidos Dinámicamente (DSO) está recomendado pero no es obligatorio.

Apache.

El software del Servidor Apache está disponible en el sitio Web del grupo Apache y en decenas de sitios *mirror* de todo el mundo. Trate de encontrar un sitio *mirror* que le sea próximo geográficamente. Evidentemente, la proximidad geográfica no significa necesariamente que un sitio esté próximo en territorio de conectividad de red, pero supone un buen punto de partida.

Los URL importantes son:

- La fundación Apache Software, que está en http:/www.apache.org/. hay más de un proyecto bajo la protección de la ASF, aunque Servidor Apache es el más conocido.
- El Proyecto Apache HTTP Server, en http://www.apache.org/httpd.html. este sitio es la fuente de información más exacta y actualizada que existe sobre Servidor Apache. Toda la documentación del servidor está disponible en línea, así y otros tipos de recursos relacionados con Apache.
- Descargar Apache en http:/www.apache.org/dist/. Esta es la ubicación principal para obtener el código fuente de Apache.
- Los *mirror* del proyecto Apache en http:/www.apache.org/mirrors/. Este sitio enumera, por código de país, los sitio *mirror* oficiales de Apache.

Apache esta disponible para su descarga en varias versiones, en código binario y en código fuente, en la página de descarga de Apache y en los distintos *mirror*. Ciertas personas prefieren utiliza una versión más antigua del producto, porque con ello saben que el software que están usando está probado. El uso de la última versión del producto no está destinado a todo el mundo, y pone nerviosos a los administradores de sistemas, especialmente si se juega el puesto de trabajo.

Descargar la última versión siempre es lo más seguro, ya que el grupo apache prueba el software antes de que se pueda descargar. Sin embargo, deberá leer la lista de fallos conocidos, para así poder estar al tanto de los temas problemáticos con el software y para evitar una versión que pueda tener un problema que le afecte directamente. Para ver los temas abiertos sobre una determinada versión, ingrese a la página http://www.apache.org/bug_report.html.

Descarga Binaria.

Apache está disponible en forma binaria en una serie de plataformas. Antes de usar un binario, asegúrese de que ha sido construido con las opciones que le interese utilizar. Si desea una construcción genérica sin módulos opcionales, probablemente convenga utilizar uno de estos. Asegúrese de que lo construye con una configuración que coincida con la suya, para no encontrar así problemas de compatibilidad. Siempre es mejor construir Apache a partir del código fuente.

***Descargar Código Fuente*.**

Si va a construir Apache desde el código fuente, descargue el archivo tar.gz correspondiente a la versión que haya decidido utilizar. Por ejemplo, la última versión a fecha de esta publicación es Apache 1.3.9 por lo que el archivo que se desea descargar se denomina apache_1.3.9.tar.gz.

Verificar la Autenticidad del Archivo.

Si tiene instalado buena privacidad, podrá verificar que el archivo que ha descargado es el artículo genuino. Esto es especialmente importante si descarga el archivo desde un sitio mirror en vez de un sitio web Apache.

Verá que hay un archivo en el directorio de descarga que tiene el mismo nombre que el archivo que acaba de descargar con una extensión de archivo .asc adicional. Es la firma PGP que va con el archivo que ha descargado. Puede utilizar este archivo para verificar su autenticidad. Las claves PGP de los distintos desarrolladores Apache están en el archivo llamado KEYS, que se encuentra en este mismo directorio.

3.1.3. DIRECTIVAS DE CONFIGURACIÓN

Cuando se ha instalado el servidor, hay que configurarlo. Apache incluye por defecto una configuración que arranca el servidor en el puerto TCP por defecto, que es el puerto 80, y sirve los ficheros del directorio que se ha especificado mediante la directiva de configuración denominada DocumentRoot. Este fichero de configuración de Apache es el httpd.conf, localizado en el subdirectorio "conf" dentro del directorio de instalación. httpd.conf es un fichero de tipo ASCII que contiene las directivas de configuración.

Estructura del Fichero de Configuración Httpd.Conf

httpd.conf está compuesto por tres bloques fundamentales, aunque las directivas de cada uno de los bloque pueden aparecer desordenadas o mezcladas.

Estos bloques son:

- Parámetros globales
- Directivas de funcionamiento
- Hosts virtuales

Algunos parámetros son de propósito general, y otros son configurables de forma independiente para cada conjunto de directorios o de ficheros o incluso para un servidor virtual específico. En tales casos, estos parámetros se encuentran dentro de secciones en las que se indica el contexto de aplicación de dicho parámetro. Las secciones fundamentales son:

<Directory>: Los parámetros que se encuentran dentro de la sección Directory sólo se aplican al directorio indicado y sus subdirectorios.

<DirectoryMatch>: Igual que Directory, aunque acepta expresiones regulares en el nombre del directorio.

<Files>: Los parámetros de configuración facilitan control de acceso a los ficheros mediante su nombre.

<FilesMatch>: Igual que Files, pero acepta en el nombre del fichero expresiones regulares.

<Location>: Proporciona control de acceso a los ficheros mediante la URL.

<LocationMatch>: Igual que Location, pero acepta en el nombre del fichero el uso de expresiones regulares.

<VirtualHost>: Los parámetros sólo se aplican a aquellas peticiones dirigidas a este host (nombre de servidor, dirección IP o puerto TCP).

<Proxy>: Sólo se aplican estos parámetros a aquellas peticiones de proxy (requiere que esté instalado "***mod proxy***") coincidentes con la especificación de URL.

<ProxyMatch>: Igual que proxy, pero acepta en la URL indicada el uso de expresiones regulares.

<IfDefine>: Sólo se aplica si al arrancar el servidor existe un parámetro concreto definido en la línea de comandos, mediante la ***opción -D***.

<IfModule>: Se aplican los parámetros si el módulo especificado se encuentra cargado (mediante LoadModule) en el momento de arrancar el servidor.

En caso de que exista conflicto entre diferentes especificaciones de parámetros, el orden de precedencia es:

1. <Directory> y .htaccess
2. <DirectoryMatch> y <Directory>
3. <Files> y <FilesMatch>
4. <Location> y <LocationMatch>

En cuanto a <VirtualHost>, estas directivas siempre se aplican después de las directivas generales. De este modo, un VirtualHost puede modificar la configuración por defecto.

Un ejemplo de configuración:

```
<Directory /home/*/public_html>
  Options Indexes
</Directory>
<FilesMatch \.(?i:gif jpe?g png)$>
  Order allow,deny
  Deny from all
</FilesMatch>.
```

Directivas Globales de Configuración

Algunas directivas de configuración nunca se aplican a las secciones antes mencionadas (directorios, etc.), sino que afectan al conjunto del servidor web. Las más destacables son:

ServerRoot: especifica la localización del directorio raíz en el que se encuentra instalado el servidor web. Partiendo de este directorio, se encuentran los ficheros de

configuración, etc. Si la instalación del servidor es correcta, no debería modificarse nunca.

KeepAlive: especifica si se deben utilizar conexiones persistentes para atender las peticiones de un mismo usuario mediante la misma conexión TCP.

Listen: especifica el puerto en que se atenderán las peticiones. Por defecto el servidor "escucha" en el puerto 80 de TCP. Permite especificar las direcciones IP que se utilizarán (en caso de que el servidor tuviese más de una). Por defecto se utilizarán todas las disponibles.

LoadModule: Permite cargar en el servidor los módulos adicionales de Apache. La sintaxis es:

LoadModule modulo ficheromodulo

Se debe tener instalado mod_so para poder utilizarla.

Directivas Principales

Hay algunas directivas que, generalmente, no suelen aparecer en las secciones anteriormente mencionadas (algunas de ellas no deben estar en ninguna sección, y es obligatorio que aparezcan en la sección principal), sino que se encuentran en la sección principal. Estas directivas son:

ServerAdmin: especifica la dirección de correo electrónico del administrador. Esta dirección puede mostrarse en los mensajes de error a modo de dirección de contacto para que los usuarios notifiquen el error al administrador. No debe estar dentro de ninguna sección.

ServerName: sirve para especificar el nombre y el puerto TCP que el Apache utiliza para identificarse. Se puede determinar de forma automática, pero se recomienda especificarlo. Si el servidor no tuviera un nombre DNS, es recomendable incluir su dirección IP. No debe incluirse dentro de ninguna sección. Su sintaxis es:

ServerName nombredireccion: puerto

Como en:

ServerName www.uoc.edu:80
ServerName 192.168.1.1:80

DocumentRoot: directorio raíz desde el cual se servirán los documentos. Por defecto es "htdocs", dentro de la carpeta de instalación de Apache. No debe aparecer dentro de ninguna sección, a excepción de la sección VirtualHost. Le corresponde una sección <Directory> en la cual se marcan los parámetros de configuración de este directorio.

DirectoryIndex: especifica el fichero que Apache servirá por defecto para cada directorio en caso de que no se especifique ningún fichero concreto en la URL de la petición. Por defecto es "index.html". Es decir, si se solicita en la barra de direcciones del navegador: *www.cibernetia.com* el servidor enviará por defecto *www.cibernetia.com/index.html*. Es posible especificar más de un fichero y el orden con que se especifican los ficheros determinará la prioridad para determinar cuál se debe servir. Es posible encontrar la directiva fuera de cualquier sección o dentro de alguna de ellas.

AccessFileName: determina el fichero de configuración en caso de que éste no sea .htaccess. Para que esta configuración cumpla su cometido, la directiva AllowOverride debe tener un valor adecuado. No puede estar incluida en ninguna sección. El fichero por defecto es .htaccess.

ErrorDocument: esta directiva establece el comportamiento de Apache en caso de error. Existen 4 configuraciones distintas:

- Mostrar algún texto de error.
- Redirección hacia un fichero en el mismo directorio.
- Redirección hacia un fichero en nuestro servidor.
- Redirección hacia un fichero fuera de nuestro servidor.

La sintaxis es:

ErrorDocument código_error acción.

Es posible encontrar esta directiva tanto dentro de una sección, como en la configuración global.

Por ejemplo:

ErrorDocument 404 /noencont.html.

En caso de que Apache no encuentre un fichero, se mostrará el fichero noencont.html.

Alias: las directivas Alias y AliasMatch permiten la definición de accesos a directorios que están fuera del DocumentRoot. Su sintaxis es:

Alias url directorio.

Por ejemplo:

Alias /docs / home/documentos

Hará que una petición a http://www.uoc.edu/docs/manual se sirva desde /home/documentos/manual.

UserDir: permite indicar a Apache que un subdirectorio dentro del directorio de trabajo de los diferentes usuarios del sistema sirva para que estos almacenen su página personal.

Por ejemplo:

UserDir publico

Hará que la página almacenada en el directorio del usuario "test", dentro del subdirectorio "público", sea accesible como:

http://www.uoc.edu/~test/indice.html

Directivas De Sección

Casi todas las secciones de localización (Directory, Location, etc.) incluyen una serie de directivas en su configuración que permiten controlar el acceso al contenido. El módulo mod_access facilita estas directivas.

Allow: permite especificar quién tiene autorización para acceder a un recurso. Se pueden especificar direcciones IP, nombres de máquina, fragmentos del nombre o de la dirección o variables de la petición. Existe la palabra clave "all" que indica "todos los clientes".

Deny: permite especificar a quién no permitimos el acceso a un recurso. Cuenta con las mismas opciones que Allow.

Order: permite afinar el funcionamiento de las anteriores directivas: Allow y Deny. Existen 2 opciones:

- ***Allow,Deny***. Por defecto se deniega el acceso y sólo podrán acceder aquellos clientes que cumplan las especificaciones de Allow y en cambio no cumplan las especificaciones de Deny.
- ***Deny,Allow***. Por defecto se permite el acceso y sólo podrán entrar los clientes que no cumplan las especificaciones de Deny o sí cumplan las especificaciones de Allow.

Servidores Virtuales

Apache permite servir varios sitios web con un único servidor. Para ello permite la creación de dominios virtuales en función de diferentes direcciones IP o diferentes nombres por IP. Apache fue de los primeros servidores que soportó servidores virtuales sin necesidad de distinguir por IP, sino en función de nombre. Esta capacidad simplifica enormemente la administración de los servidores, y supone un ahorro de direcciones IP, que normalmente son escasas. Los servidores virtuales que distinguen en función del nombre son perfectamente transparentes para el cliente, con la posible excepción de aquellos navegadores muy antiguos que no envíen la cabecera "Host:" con cada petición.

Servidores Virtuales Por Dirección Ip

Para atender a varios servidores virtuales, cada uno de ellos con una dirección IP diferente, se utiliza la sección de configuración ***VirtualHost***. Con esta sección se define una configuración y dirección IP para cada uno de los servidores.

Un ejemplo sería el siguiente.

```
<VirtualHost 192.168.1.1> enter  ServerAdmin webmaster@uoc.edu
  DocumentRoot /web/uoc
  ServerName www.uoc.edu
  ErrorLog /web/logs/uoc_error_log
  TransferLog /web/logs/uoc_access_log
</VirtualHost>
```

```
<VirtualHost 192.168.254.254>
  ServerAdmin webmaster@asociados.uoc.edu
  DocumentRoot /web/asociados
  ServerName asociados.uoc.edu
  ErrorLog /web/logs/asociados_error_log
  TransferLog /web/logs/asociados_access_log
</VirtualHost>
```

Este ejemplo define 2 servidores web, cada uno de ellos con una IP y un nombre diferentes. Ambos tienen su propio DocumentRoot, etc.

Para hacer uso de servidores virtuales por IP, se necesita que el sistema servidor tenga configuradas en su sistema operativo las diferentes direcciones IP que debe servir.

Servidores Virtuales Por Nombre

Para atender a varios servidores, utilizando una misma dirección IP para todos ellos, se utiliza la sección ***VirtualHost***, que permite definir los parámetros de cada uno de los servidores. Sirva como ejemplo la siguiente configuración:

```
NameVirtualHost *:80
  <VirtualHost *:80>
    ServerAdmin webmaster@uoc.edu
    ServerName www.uoc.edu
    DocumentRoot /web/uoc
    ErrorLog /web/logs/uoc_error_log
    TransferLog /web/logs/uoc_access_log
  </VirtualHost>
  <VirtualHost *:80>
    ServerAdmin webmaster@uoc.edu
    ServerName asociados.uoc.edu
    DocumentRoot /web/asociados
    ErrorLog /web/logs/asociados_error_log
    TransferLog /web/logs/asociados_access_log
  </VirtualHost>.
```

Se puede utilizar una dirección IP concreta en lugar de *, lo cual permite asignar, por ejemplo, un grupo de servidores virtuales por nombre a esta IP y otro grupo a otra IP.

Cuando nuestro servidor tiene 2 direcciones IP, pero hemos asignado a las 2 el mismo nombre, se necesita un uso especial de las directivas de servidores por nombre. Por ejemplo, cuando se dispone una conexión de red en la intranet y otra conexión diferente en Internet con el mismo nombre, caso en el cual podemos servir el mismo contenido de la esta forma:

```
NameVirtualHost 192.168.1.1
NameVirtualHost 172.40.30.40
<VirtualHost 192.168.1.1 172.40.30.40>
  DocumentRoot /www/servidor1
  ServerName servidor.uoc.edu
  ServerAlias servidor
</VirtualHost>
```

Con esta configuración se puede servir la misma web hacia la intranet y hacia la Internet. Es conveniente señalar el uso de un alias para el servidor, lo cual permite no tener que usar dominios en la intranet. Disponemos de una especificación de servidor virtual por defecto "_default_" que permite atender las peticiones que no sirve ningún otro servidor virtual.

```
<VirtualHost _default_>
  DocumentRoot /www/defecto
</VirtualHost>
```

Podemos usar la etiqueta "_default_" indicando un número de puerto para especificar servidores por defecto que sean diferentes para cada puerto. Apache permite también configuraciones más complejas de servidores virtuales, muy útiles en casos de, por ejemplo, servidores masivos. Una excelente guía de referencia se encuentra en la web del proyecto Apache, con consejos útiles para configurar el servidor. Existen muchos servidores HTTP de código libre, pero la mayoría de ellos han quedado eclipsados por Apache. Algunos de estos servidores tienen características que les hacen especialmente interesantes.

3.1.4 SEGURIDAD

La World Wide Web es la frontera más grande que hay. Ha sido característico de otras fronteras que abunden las oportunidades para los ciudadanos firmes, trabajadores y sólidos (y para aquellos que viven a costa de ésto). Sólo se necesita el diario para ver

que esto sucede en la Web como sucedió en los territorios del Oeste de los Estados Unidos en el siglo XIX. Casi a diario, las noticias informan de cómo hay sitios web corporativos que se derrumban.

La "seguridad" incluye las técnicas y acciones planificadas para evitar ser victima de la nueva frontera. Desafortunadamente, la seguridad es uno de esos costes comerciales que no muestran valor positivo alguno cuando se aplican con éxito (lo único que no tiene valor negativo). Como consecuencia de ello, presupuesto para la implementación de las medidas de seguridad informática desarrolló una forma novedosa de tratar este tema. Le mostró a su jefe un titular del periódico particularmente alarmante sobre la empresa, preguntándole, "¿Cuánto vale para las empresas que no estamos en los titulares como este?"

Independientemente de los métodos que se empleen para proteger el sistema web, hay que estar alertas frente a las dos últimas amenazas: el acceso no autorizado a la información a través de la Web y el acceso no autorizado al propio sistema web.

***Proteger los Archivos del Servidor Web*.**

De hecho, casi todos los documentos web se basan en archivos de un disco que está en el propio servidor web. Por tanto, el hecho de modificar o dañar los archivos subyacentes modificará o dañará las páginas tal como se ven en la web, además, los distintos controles de seguridad operativos del software del servidor web están representados en términos de archivos de disco, por lo que proteger los archivos del servidor es una preocupación elemental en la base de todos los mensajes de seguridad.

***Archivos de Lectura-Escritura frente a archivos de sólo lectura*.**

La mayoría de archivos que controlan el funcionamiento del servidor web se encuentran en el árbol de directorio de ***ServerRoot***. Existen excepciones, como los archivos .htaccess y, posiblemente, bases de datos de autorización, pero estos sólo afectan a los aspectos secundarios del comportamiento del servidor.

Cuando se ejecuta el servidor, lo hace con una serie de derechos de accesos que están asociados con algún ID de usuario del sistema, como **root**, **nobody** o, posiblemente httpd en sistemas Unix y, normalmente, con **LocalSystem** en Windows NT. Cuando

nos referimos al servidor en los párrafos siguientes, significa “la identificación con la que se ejecuta el servidor”.

El servidor, por regla general, no necesita (y no debería tener) la capacidad de modificar ninguno de sus archivos de control. Las excepciones más obvias son el registro de error y los archivos de registro de acceso, que necesita el servidor para grabar información a medida que se van produciendo los eventos. Tener archivos como el archivo de configuración principal, http.conf, donde el servidor puede escribir, es una situación potencialmente desastrosa, ya que abre la posibilidad de que una mala configuración permita que un intruso en el web arruine el servidor y lo use para modificar su propia configuración.

Con la excepción de los archivos de registro (que puede que ni siquiera estén bajo el árbol **ServerRoot**, si ha usado las directivas **ErrorLog** y **CustomLog** para ponerlas en otro sitio), ninguno de los archivos de **ServerRoot** deberán ser modificables por el propio servidor, sino sólo por la persona (o personas) responsables de gestionar el servidor.

De manera análoga, **DocumentRoot** debe ser de sólo lectura en lo que respecta al servidor. Sin embargo, esta regla no es estricta, ya que los entornos específicos del sitio (como el uso de WebDAV para permitir la gestión remota de documentos) podrían imponer que la modificabilidad es un atributo necesario de algunos documentos.

La regla básica es esta: ¿desea que un extraño sea capaz de alterar este archivo desde la Web? Si la respuesta es negativa, los permisos y propiedades del archivo no deberán permitir que el servidor pueda modificarlo.

***Proteger los URL en los Sitios Web*.**

Si presuponemos que todos los archivos del servidor se encuentran protegidos apropiadamente, ¿qué pasa con los documentos web que son visibles a través de los navegadores? ¿Pueden ojos ajenos ver balances de las empresas? ¿Puede alguien que no esté en el departamento de personal acceder a las nóminas de los empleados?

Este tema se diferencia de proteger los archivos del sistema. En algunos casos, incluso un nombre de archivo puede ser importante; si lo ha protegido de forma que nadie puede acceder a él, su mera existencia puede resultar indicativa. Por ejemplo, conocer

el nombre del archivo **business-plan.html** de la sección puede dar al intruso en blanco muy definido.

La protección de los URL no supone que se divulgue tanta información o, al menos, restringirla a la gente de confianza.

***Protección Basada en Direcciones*.**

Apache 1.3.9 proporciona una forma de limitar el acceso en base a la dirección IP del cliente. Definido por el módulo **mod_access**, este método permite a los *webmasters* autorizar o denegar el acceso a los recursos web de sus sistemas en función de la dirección que el cliente está utilizando o la red de la que procede.

Dado que la dirección del cliente se conoce cuando realiza una conexión con el servidor, la decisión de si hay que dejarle que continúe puede hacerse en forma muy rápida.

***Utilizar Directivas Allow y Deny*.**

El módulo **mod_access** permite a los *webmasters* especificar de que *host* o red IP va a permitir el servidor que provengan las solicitudes. Las reglas básicas funcionan por exclusión, como "autorizar a todo el mundo, a excepción de esta red" o "negar el acceso a todo el mundo, a excepción de las personas de este sistema".

De hecho, las dos directivas principales necesarias para gestionar este nivel de control de acceso son ***allow*** y ***deny***. La directiva allow específica los criterios para permitir el acceso, mientras que deny indica las reglas para desautorizarlo. En estas directivas existen dos formas:

- Allow | deny from address-expression
- Allow | deny from env=environment-variable

Address-expression pueden ser:

- La palabra clave especial **all**, lo que significa que todos los hosts posibles queden afectados.
- Un nombre de host o de dominio parcial o total, como hoohoo.ncsa.uiuc.edu o .ncsa.uiuc.edu

- Un dirección IP completa, como 127.0.0.1
- Una dirección parcial, como 10.0.0
- Un par de red máscara de la red, como 10.0.0.0/255.255.0.0
- Una especificación de dirección CIDR, como 127.0.0.0/24 (es la misma que 127.0.0.0/255.255.255.0)

Por motivo de rendimiento, se recomienda que utilice direcciones reales en vez de nombres de dominios o de host. Cuando se realiza una conexión con el servidor con el servidor, Apache conocerá la dirección IP del cliente; no obstante para verificar el nombre de host, tendrá que traducir la dirección en un nombre por razones de comparación.

Puede limitar el acceso a las páginas con combinaciones de estas directivas, como con:

Deny from all

Allow from 127.0.0.1

O *Allow from all*

Deny from spamhost.org

***Order*.**

Dado que las directivas **allow y deny** se complementan entre sí se usan conjuntamente, tiene que haber una forma de indicar qué se deben procesar primero. Esto lo proporciona la directiva **Order**, que debe preceder a cualquier directiva **allow** o **deny** de un determinado alcance y que puede presentarse en uno de los siguientes formatos:

- Order allow, deny
- Order deny, allow
- Order mutual-failure

Los significados de los dos primeros formatos se explican por sí solos. **Order Allow, deny** incluye a Apache que compruebe todas las instrucciones **Allow** antes de ver las directivas **Deny**, y lo contrario en **Order deny, allow**.

Las palabras claves **mutual-failure** necesitan una explicación más profunda. Cuando se usan, para que se permita una solicitud, el host que la realiza debe cumplir al menos una condición **allow** y no hacer que se cumpla ninguna **deny**. Esto significa

que **mutual-failure** no funciona en todas las configuraciones excluyentes utilizando **deny from all.**

¿Cuál es el acceso predeterminado que se autoriza a un cliente que no cumple las condiciones? En lo que respecta a la orden **allow, deny**, por defecto, cada cliente empieza siendo rechazado; no será autorizado hasta que cumpla al menos una de las condiciones **allow.** En un enlace cubierto por **Order deny, allow,** la condición inicial consiste en permitir todo el acceso hasta que la desactive una directiva **deny.**

La forma más sencilla de imaginarse el estado inicial consiste en recordar que coincide con la última palabra clave del orden. En **Order allow, deny** se niega; en **Order deny, allow** se autoriza.

No existe estado inicial para el modo **mutual-failure**. Para tener acceso, un cliente debe ser permitido y no debe ser rechazado.

Credenciales Facilitadas por el Usuario

Cuando un cliente (normalmente un navegador) trata de acceder a un documento protegido por controles discrecionales, el servidor Apache responderá con un estado de error (404, Authorization_Required).

Control de la Autorización

La autenticación se maneja de forma muy sencilla en Apache: las credenciales que somete el cliente son válidas o no lo son. Sin embargo, la autenticación y la autorización van muy unidas. Si un recurso no está bajo ningún tipo de control de acceso, apache no comprobará las credenciales.

Dado que el control de acceso obligatorio utiliza las credenciales inherentes a la propia solicitud y no depende de nada de lo que envía el usuario, siempre en los alcances donde se colocan los controles necesarios. Los controles discrecionales necesitan ser activados (básicamente, necesita indicarle al servidor que “compruebe las credenciales y las autorizaciones de esta esfera”). Esto se hace con la directiva **Require**.

Require.

La directiva **Requiere** es la que permite la comprobación del acceso discrecional. Una esfera puede tener un nombre (como AuthName “Business Plan”) y un método de

autenticación (como AuthType Basic) e, incluso, una base de datos de autenticación definida, pero si no existe la directiva Require, ninguno de los otros surtirá efecto.

La palabra clave y los argumentos de la directiva Require son completamente arbitrarios en lo que respecta al servidor Apache central. Solo se graba la información, observa que es necesaria la autorización discrecional en el enlace actual y hace que los argumentos estén a disposición de los métodos cuando lo piden. Sin embargo, he aquí los valores significativos del módulo **mod_auth** que se incluyen en el paquete básico de Apache:

- *User username username…*- El acceso está organizado si la parte nombre de las credenciales que entrega el cliente coinciden con alguno de los nombres de usuarios especificados y si el par nombre de usuario-contraseña está autenticado en el archivo **AuthUserFile.**
- *Valid-user.*- El acceso está garantizado si el nombre de usuario y contraseña coinciden con los archivos **AuthUserFile.**
- *Group groupname groupname….*- El acceso está garantizado si el nombre de usuario y la contraseña se autentican con éxito y el nombre de usuario aparece en el archivo **AuthGroupFile** como que está en uno de los grupos específicos.

La condición previa común a todos estos valores es que el nombre de usuario y la contraseña que da el usuario como parte de la solicitud deben ser válidos y aparecer en la lista de credenciales **AuthUserFile.**

Dado que cada módulo de autenticación interpreta el parámetro **Require** de forma propia, algunos pueden exigir que todas las condiciones coincidan, mientras que los demás permiten acceso solamente si uno lo permite. El módulo **mod_auth** recae en la última categoría; las condiciones de la primera directiva **Require** del alcance deben ser satisfechas para que se permita el acceso.

Controlar La Actividad En Tiempo Real

Las secciones anteriores trataron sobre la colocación de los controles de accesos en los documentos y sobre el control de quien o que podría acceder a ellos. Sin embargo, otros aspectos del funcionamiento de Apache tienen repercusiones relativas a la

seguridad del sistema, como el tipo de scripts que se permite ejecutar y bajo que circunstancias.

***Opciones y omisiones*.**

Cada alcance posee una lista de opciones que se activan y una lista de los tipos de directivas que pueden aparecer en los archivos .htaccess dentro del alcance. La primera lista la maneja la directiva **Options**, que puede adoptar una o más de las siguientes palabras claves:

- **All** permite que todas las opciones, e excepción de **MultiViews** (que debe estar activada, como ocurre con **Options All MultiViews**). Este es el parámetro predeterminado de un alcance en caso de que no aparezcan directivas **Options** en un alcance más alto o amplio.
- **ExecCGI** permite la ejecución de scripts CGI dentro del alcance.
- **FollowSymLinks**, en caso de que esté activado, hace que el servidor siga los vínculos simbólicos que encuentre dentro del alcance. Dado que un symlink es un concepto de sistema de archivos, en un contenedor <Location> esta opción no tiene significación y se omite.
- **Includes** activa el procedimiento de la inclusión del lado del servidor por parte de **mod_include** dentro del alcance en los archivos que opten por ellos.
- **IncludeNoEXEC** es parecido al includes, con la excepción de que desactiva la directiva de inclusión del lado del servidor #exec de mod_include y el uso de la directiva SSI #Include para incluir la salida de los scripts CGI.
- **Indexes** controla si usa mod_autoindex para general listados de directorios predeterminados.
- **MultiViews** permite la negociación de contenido para el alcance, permitiendo que el servidor deduzca el nombre del documento adecuado o el mejor de una lista de opciones posibles que coincidan con los criterios solicitados.
- **SymLinksslfOwnerMatch** es similar a la opción FollowSymLinks, con la excepción de que los vínculos simbólicos sólo se siguen si el propietario del vínculo es el mismo que el propietario del documento al que señala el vínculo.

Cada una de estas palabras claves, cuando están activadas, puede suponer información nueva o distinta que se muestra en respuesta a una solicitud. También

puede suponer que los usuarios de su propio servidor web violen sin darse cuenta la seguridad al instalar un scripts poco implementado o un vínculo no intencional.

Servidor Web Enjaulado

El siguiente paso es limitar a los procesos de Apache a que accedan a los sistemas de ficheros. La técnica del enjaulado, es como se describe a continuación:

Código:

```
mkdir -p /chroot/httpd/dev
mkdir -p /chroot/httpd/etc
mkdir -p /chroot/httpd/var/run
mkdir -p /chroot/httpd/usr/lib
mkdir –p /chroot/httpd/usr/libexec
mkdir –p /chroot/httpd/usr/local/apache2/bin
mkdir -p /chroot/httpd/usr/local/apache2/lib
mkdir -p /chroot/httpd/usr/local/apache2/logs/www.ebank.lab
mkdir -p /chroot/httpd/usr/local/apache2/logs/www.test.lab
mkdir -p /chroot/httpd/usr/local/apache2/conf
mkdir -p /chroot/httpd/usr/local/lib
mkdir -p /chroot/httpd/www
```

El dueño de los directorios anteriores debe ser el usuario root, y los derechos de acceso no deberían permitir que usuarios normales hagan cambios en dichos directorios:

Código:

```
chown -R root:sys /chroot/httpd
chmod -R 0755 /chroot/httpd
```

Ahora crearemos, si no lo está ya, el dispositivo especial, /dev/null.

Código:

```
ls -al /dev/null
  crw-rw-rw- 1 root wheel 2, 2 Mar 14 12:53 /dev/null
mknod /chroot/httpd/dev/null c 2 2
chown root:sys /chroot/httpd/dev/null
chmod 666 /chroot/httpd/dev/null
```

También necesitamos crear un dispositivo /chroot/httpd/dev/log que será necesario

para que el servidor trabaje correctamente. En el caso de nuestro sistema FreeBSD, la siguiente línea debería añadirse a /etc/rc.conf:

Código:

```
syslogd_flags="-l /chroot/httpd/dev/log"
```

Para que los cambios surtan efecto, necesitamos también reiniciar el demonio syslogd con el nuevo parámetro:

Código:

```
kill  cat /var/run/syslog.pid
/usr/sbin/syslogd -ss -l /chroot/httpd/dev/log
```

El siguiente paso es copiar todos los programas, librerías y archivos de configuración necesarios en el nuevo árbol de directorios. Para el caso de FreeBSD 4.1 la lista de los archivos requeridos es la siguiente:

Código:

```
cp /usr/local/apache2/bin/httpd  /chroot/httpd/usr/local/apache2/bin/
cp /usr/local/apache2/lib/libaprutil-0.so.9    /chroot/httpd/usr/local/apache2/lib/
cp /usr/local/apache2/lib/libapr-0.so.9  /chroot/httpd/usr/local/apache2/lib/
cp /usr/local/apache2/conf/mime.types  /chroot/httpd/usr/local/apache2/conf/
cp /usr/local/apache2/conf/httpd.conf  /chroot/httpd/usr/local/apache2/conf/
cp /usr/local/lib/libexpat.so.4  /chroot/httpd/usr/local/lib/
cp /usr/lib/libc.so.5  /chroot/httpd/usr/lib/
cp /usr/lib/libcrypt.so.2  /chroot/httpd/usr/lib/
cp /usr/lib/libm.so.2  /chroot/httpd/usr/lib/
cp /usr/libexec/ld-elf.so.1  /chroot/httpd/usr/libexec/
cp /var/run/ld-elf.so.hints  /chroot/httpd/var/run/
cp /etc/hosts  /chroot/httpd/etc/
cp /etc/nsswitch.conf  /chroot/httpd/etc/
cp /etc/resolv.conf  /chroot/httpd/etc/
cp /etc/group  /chroot/httpd/etc/
cp /etc/master.passwd  /chroot/httpd/etc/passwords
```

Para el caso de otros sistemas Linux, Unix y BSD, la lista de los archivos necesarios puede determinarse usando los comandos ldd, strace, truss o strings.

Después de que los pasos anteriores están completados, necesitamos preparar la base de datos de passwords que debe haber en el presente sistema enjaulado. Así, de /chroot/httpd/etc/passwords y /chroot/httpd/etc/group tenemos que quitar todas

las líneas, excepto apache. Ahora, deberíamos construir la base de datos de passwords como sigue:

Código:

```
cd /chroot/httpd/etc
pwd_mkdb -d /chroot/httpd/etc passwords
rm -rf /chroot/httpd/etc/master.passwd
```

Estos comandos deben ejecutarse si se usa FreeBSD. En otros sistemas, podría ser suficiente con editar los archivos /chroot/httpd/etc/passwd y /chroot/httpd/etc/shadow.

Finalmente, podemos copiar el contenido de ejemplo del sitio web al entorno enjaulado:

Código:

```
cp -R /www/* /chroot/httpd/www/
```

Y comprobar que el servidor Apache funciona correctamente:

Código:

```
chroot /chroot/httpd /usr/local/apache2/bin/httpd
```

Pasos finales.

Si tu Apache ahora funciona correctamente, lo único que nos queda es crear un script que ejecute Apache en tiempo de arranque. Para hacer esto, puede utilizarse el script apache.sh:

Código:

```
#!/bin/sh
CHROOT=/chroot/httpd
HTTPD=/usr/local/apache2/bin/httpd
PIDFILE=/usr/local/apache2/logs/httpd.pid

echo -n " apache"

case "$1" in
```

```
start)
    /usr/sbin/chroot $CHROOT $HTTPD
    ;;
stop)
    kill `cat ${CHROOT}/${PIDFILE}`
    ;;
*)
    echo ""
    echo "Usage: `basename $0` {start|stop}" >&2
    exit 64
    ;;
esac
exit 0
```

El anterior script debería copiarse al directorio donde los scripts de arranque están ubicados. Para el caso de FreeBSD es /usr/local/etc/rc.d. Los derechos de acceso que el fichero debe contener deberían ser:

Código:

```
chown root:sys /usr/local/etc/rc.d/apache.sh
chmod 711 /usr/local/etc/rc.d/apache.sh
```

3.1.5. CARACTERÍSTICAS DE CONFIGURACIÓN

La cantidad de funciones y servicios que ofrece el servidor web Apache es enorme, y además puedes incorporar nuevos módulos de servicios de acuerdo a tus necesidades específicas. Te puede resultar muy útil, tener la **cartilla de las directivas** del servidor Apache, te va a facilitar las tareas de configuración y mantenimiento de tu sistema.

Hospedaje Virtual

Un solo computador ejecutando el servidor web Apache, puede actuar como si fueran muchos servidores independientes en Internet. Significa que el servidor puede manejar varias direcciones IP de Internet simultáneamente. Cada dirección tiene su página web asociada. Se denomina hospedaje múltiple, hospedaje virtual o servidor virtual.

La directiva **VirtualHost** permite definir los servidores virtuales. Se deben especificar

en el archivo de configuración **httpd.conf**. El siguiente ejemplo muestra la definición de dos servidores virtuales con direcciones IP 212.100.204.120 y 98.202.64.16 diferentes.

```
<VirtualHost        212.100.204.120        >
ServerName                    www.cursos.net
DocumentRoot                  /home1/aplice/
.....
</VirtualHost>

<VirtualHost        98.202.64.16           >
ServerName                    www.cabinas.net
DocumentRoot                  /home1/cabinas/
.....
</VirtualHost
```

Figura 9 Directiva Virtual Host

En el registro de la directiva **VirtualHost** se destacan dos datos: el nombre del servidor (ServerName) y el directorio raíz (DocumentRoot) donde se alojan los documentos, datos, archivos y páginas web, que pertenecen al servidor virtual nombrado.

A partir de la versión 1.1 el protocolo HTTP, incluye en el encabezado de requerimiento el nombre del servidor (**Host:** www.nombre.com) destino de la conexión. Ello ha posibilitado usar una misma dirección IP para definir tantos **servidores virtuales** como necesites.

El siguiente ejemplo muestra la definición de dos servidores virtuales, www.aplice.net y www.redtrans.net que usan la misma dirección IP 201.200.49.201. Cuando se comparte una dirección IP, hay que usar la directiva **NameVirtualHost** para indicar la IP común.

```
NameVirtualHost     201.200.49.201

<VirtualHost    201.200.49.201    >
ServerName          www.aplice.net
DocumentRoot        /www/a/aplice/
.....
</VirtualHost>

<VirtualHost    201.200.49.201    >
ServerName          www.redtrans.net
DocumentRoot        /www/r/redtrans/
.....
</VirtualHost>
```

Figura 10 Directiva NameVirtualHost

Funcionamiento de los servidores virtuales Al arrancar el servidor web Apache, lee el archivo **htppd.conf** con la especificación de los servidores virtuales, y crea una tabla con los registros de éstos, ordenada por las direcciones IP de los mismos.

IP	Nombre	Descripción
98.202.64.16	----------	servidor virtual con IP propia
201.200.49.201	--> www.aplice.net	servidor virtual con IP compartida
	--> www.redtrans.net	
212.100.204.120	----------	servidor virtual con IP propia

Tabla 10 Tabla de Servidores Virtuales

Cuando un cliente web, solicita una página en un servidor web, ensambla un paquete de datos que contiene la dirección IP destino, el nombre del servidor destino y el nombre de la página. Luego envía el bloque de datos a través de Internet hacia el servidor destino. Supongamos que el cliente web, solicita la página **www.redtrans.net/frances.htm**, el bloque ensamblado tendrá los datos que se muestran en el esquema siguiente.

Cuando el cliente web, recibe el bloque de datos del servidor, analiza el encabezado y se entera de la cantidad de bytes o longitud del segmento de datos enviado por el servidor. En el ejemplo, el archivo de la página web pedida tiene una longitud de 24568 caracteres.
Seguidamente el cliente web, obtiene los datos y los procesa de acuerdo a su objetivo. Por ejemplo, si el cliente web es tu navegador de Internet, despliega en tu pantalla la página web que has solicitado.

Páginas Web Dinámicas

Las páginas web, son documentos de hipertexto, en lenguaje HTML. Por definición, el formato y el texto de un documento en lenguaje HTML (Hyper Text Markup Language) son estáticos o invariables. Todos los usuarios perciben el mismo documento.

Para poder brindar un mejor servicio a los usuarios, se han desarrollado diversas técnicas y lenguajes, para desarrollar páginas web dinámicas. Un ejemplo muy típico lo ofrece el comercio electrónico. A medida que el usuario va eligiendo productos y los va ubicando en su carrito de compras, la página web que muestra el detalle de lo comprado, tiene que ser configurada para cada usuario en el momento de la consulta. No podría ser estática.

El servidor web Apache puede incorporar en el momento de su compilación, los principales lenguajes de programación de páginas web dinámicas. Como consecuencia, la producción de tales páginas web es muy eficiente, rápida y confiable.

Técnicas De Inserción

Una forma de hacer dinámica una página consiste en insertar instrucciones en el texto estático HTML. Cuando la página es solicitada, el servidor explora el texto de la página, ejecuta cada instrucción y en su lugar deja el resultado. Existen potentes lenguajes de inserción de datos. El más usado con el servidor Apache es PHP. El servidor Windows 2000 tiene el ASP (Active Server Page). El Apache tiene el SSI (Server Side Include), el cual es un lenguaje nativo de inserción de datos, sencillo pero que puede ser muy útil.

En el esquema anterior, se muestra el hipertexto de una página web, que contiene una instrucción SSI que obtiene la fecha en curso. Cuando un cliente web solicita la página **ejemplo.shtml**, el servidor Apache sabe por el sufijo **.shtml**, que tiene que explorar el hipertexto por la presencia de instrucciones SSI de inserción de datos. El identificador de una instrucción SSI es **<!--#**. Cuando halla una instrucción SSI, ejecuta el comando especificado en **cmd**, y el resultado se inserta en el lugar de la instrucción.

En el ejemplo, el comando es **/bin/date**, programa del sistema Unix, que despliega la hora y fecha actual. En el cuadro siguiente, se muestra el hipertexto que resulta después de la inserción de datos producidos por las instrucciones SSI.

Es importante destacar que el hipertexto producido por el servidor, después de procesar las instrucciones SSI, es estático. La página web que se envía al cliente web es estática.

El servidor genera en forma dinámica la página web **ejemplo.shtml** pero el cliente web recibe una página de hipertexto estático, sin instrucciones SSI, las cuales no entendería.

Para aplicar la función SSI en las páginas de un directorio, es necesario establecer un conjunto de directivas apropiado. El cuadro siguiente muestra un ejemplo típico.

En este ejemplo, todas las páginas web en el directorio /www/htdocs, cuyos nombres tienen la extensión **.shtml** se consideran que incluyen instrucciones SSI y por ende son exploradas por el servidor y producidas antes de ser enviadas.

La sintaxis general de una instrucción SSI es: **<!--#** [**exec**] **cmd="programa" >**

Puede considerarse que el SSI es un lenguaje sencillo de inserción de datos. En cambio, el lenguaje de inserción PHP, muy usado en el ambiente Unix, es realmente potente. Permite el desarrollo eficiente de páginas web dinámicas muy complejas. Facilita el acceso desde una página web, a bases de datos y a los protocolos de transmisión de datos.

Interfases CGIs

Una interfase CGI (Common Gateway Interface), es un programa que se ejecuta desde una página web y es capaz de procesar datos en el servidor y generar una página web.

Cuando un usuario, desde una página web, ejecuta un programa CGI, éste procesa los datos que correspondan y crea la página web resultado, y la pasa al servidor que se encarga de enviarla al usuario. El usuario recibe una página web estática, que no tiene existencia real en el servidor.

Los lenguajes de programación más usados para implementar programas CGIs son PERL, C/C++ y Visual Basic. Sin embargo, cualquier lenguaje disponible puede ser usado para programar una interfase CGI. El sufijo **.pl** corresponde a PERL y el sufijo **.cgi** es genérico.

El lenguaje PERL 5.x orientado a objetos, tiene el ambiente de programación de CGIs más completo. Tiene una biblioteca pública de módulos para todas las aplicaciones

que son comunes. Constantemente se mejoran y agregan nuevos módulos a dicha biblioteca.

Un ejemplo típico es la consulta de grandes bases de datos, desde una página web. El siguiente esquema muestra una página web sencilla que ejecuta un programa CGI.

Cuando un usuario ingresa el código **NS16C550A** y presiona el botón de BUSCAR, ocurre la secuencia de eventos que se describen brevemente a continuación.

1) El navegador, ensambla un paquete de datos en el cual incluye el nombre del programa CGI **chipdata.pl**, y el código NS16C550A, entre otros. Seguidamente lo envía al servidor.

2) Cuando el servidor analiza el requerimiento del cliente web, percibe que **chipdata.pl** es un programa CGI. Le acondiciona las variables de su ambiente e inicia su ejecución.

3) El programa en lenguaje PERL, **chipdata.pl** comienza a ejecutar y obtiene el código NS16C550A almacenado en una variable del ambiente (cargada por el servidor). Enseguida crea el encabezado de respuesta y luego ejecuta el manejador de bases de datos y le solicita información sobre el componente NS16C550A. Cuando recibe los datos sobre el integrado, escribe el hipertexto de la página web relativa al componente y enseguida pasa el paquete de datos (encabezado + hipertexto) al servidor.

4) El servidor envía el paquete de datos al navegador que hizo el requerimiento. El navegador obtiene el hipertexto del paquete y despliega la página web con información sobre el componente NS16C550A en la pantalla del computador del usuario.

Directivas para habilitar la ejecución de CGIs. El esquema siguiente muestra la forma de activar la ejecución de programas CGIs en el directorio **/www/ejemplo/cgi-bin/**. Los programas deben tener las extensiones (sufijos) **.pl** y **.cgi**.

Puedes profundizar sobre el tema páginas dinámicas, consultando textos y manuales de PERL, ASP, PHP y sobre programación de CGIs. El tema TU PROPIA PAGINA WEB te ayuda.

Multiplicidad De Idiomas

En forma sencilla, es posible lograr que cada usuario reciba la página web que solicita en su propio idioma. Se necesita una versión de la página en cada uno de los idiomas

que se quieren brindar. Por ejemplo, sea **ayuda.htm** una página de ayuda para los usuarios. Se considera muy útil ofrecer versiones en español, inglés y francés. Para ello hay que crear las páginas **ayuda.htm.es**, **ayuda.htm.en** y **ayuda.htm.fr** en los idiomas mencionados.

El siguiente esquema muestra las directivas necesarias para ofrecer documentos en múltiples idiomas en el directorio /www/ejemplo/htdocs.

Cuando el servidor recibe la solicitud de la página **ayuda.htm**, verifica las preferencias de idiomas del navegador del usuario. En forma automática selecciona la página en el idioma preferencial del usuario y la envía como respuesta. Por ejemplo, si el usuario usa el idioma español, recibirá el contenido de **ayuda.htm.es**, o sea la versión en su propio idioma.

Control Del Acceso A Documentos

Puedes restringir el acceso a los archivos que consideres conveniente. Puedes autorizar el acceso a los archivos de un directorio, solo a los usuarios de un computador en particular o bien a un grupo de usuarios. Es posible establecer para un usuario, una clave privada de acceso a los documentos en un directorio personal.

El esquema siguiente muestra las directivas que se necesitan para autorizar el acceso al directorio /www/ejemplo/privado/ solamente a los usuarios del subdominio *.port5.com.

Por ejemplo, pueden acceder aplice.port5.com, redtrans.port5.com y otros usuarios con direcciones del tipo *.port5.com, donde * representa un nombre de subdominio.

Mensajes De Error Personalizados

Es frecuente que un usuario digite mal la dirección de una página, o bien puede haber un enlace hacia una página web que ya no existe. En estos casos el servidor despliega una página estándar de error. Sin embargo, mediante la directiva **ErrorDocument** es posible indicar una página con información apropiada para orientar mejor a los usuarios.

El control de las acciones del servidor web en un directorio, puede efectuarse en forma específica mediante el archivo **.htaccess**, que puede contener todo tipo de directivas, las cuales se aplican al directorio actual y a sus hijos.

El siguiente cuadro muestra el contenido de un archivo **.htaccess** que tiene directivas para desplegar la página web NoExiste.htm cuando hay error 400 del cliente y desplegar ServError.htm cuando se presenta un error 500 del servidor.

Se debe incluir la trayectoria completa de cada página web. En el ejemplo, NoExiste.htm y ServError.htm están en el directorio raíz, por eso aparece /NoExiste.htm y /ServError.htm.

Bases De Datos - Servicios

Las bases de datos son un recurso de computación fundamental para implementar muchos servicios de información. Brindar acceso a las bases de datos mediante una página web, facilita en forma notable a los usuarios, los cuales desde la comodidad de su hogar y en el momento oportuno pueden realizar sus investigaciones.

El número de bases de datos en Internet aumenta constantemente, porque es la forma más eficiente de organizar los grandes volúmenes de datos. Son comunes en Internet bases de datos con 1000 millones de registros, pero la mayoría tienen unos pocos miles.

Las bases de datos permiten producir en forma muy eficiente páginas web dinámicas, con la información específica que necesita el usuario.

Se requiere un programa interfase para acceder a una base de datos desde Internet. Puede ser un programa CGI escrito en lenguaje PERL u otro lenguaje. También puede emplearse un programa de inserción de datos en lenguaje PHP. Los lenguajes PERL y PHP tienen drivers para comunicarse con los principales manejadores de bases de datos.
Sigue una lista de tales manejadores: dBase, DB2, Informix, InterBase, Ingres, Michosoft SQL, mSQL, MySQL, Oracle, ODBC, PostgreSQL, Sybase y muchos otros.

El servidor Apache puede incorporar los lenguajes PERL y PHP, para una ejecución más eficiente de los programas interfases que acceden a las bases de datos. En el archivo INSTALL que viene en la versión fuente, tienes ejemplos sobre las diversas formas de incorporar PERL y PHP en el servidor Apache en la fase de su compilación.

Cada lenguaje de programación de CGIs o de inserción de datos, ejemplos PERL y PHP, tienen un driver específico para cada uno de las bases de datos, a las que le

brinda soporte. Las funciones del driver permiten consultar, actualizar, agregar y borrar registros de una base de datos desde una página web. También el administrador de una base de datos, puede efectuar sus tareas, en forma remota usando Internet.

3.1.5. TUNNING

Tunning Apache en Linux

Para obtener el máximo rendimiento del Servidor web Apache, configurar el httpd.conf de Linux/Unix.

Los creadores del Apache afirman que no está diseñado, o no es su máxima prioridad, el rendimiento, pero aún así es posible configurarlo para obtener un mejor rendimiento. Especialmente es indicado para máquinas con una buena/muy buena conexión a Internet.

Configuración mínima: 256 RAM.

Con esta nueva configuración Apache consumirá más memoria, pero si contamos con bastante memoria (1 GB o 2 GB) no hay problema.

MaxKeepAliveRequests 0

 # en comentarios los valores originales

#MinSpareServers 5

#MaxSpareServers 10

MinSpareServers 16
MaxSpareServers 30

También podemos poner (pero mejor tener una máquina con un tráfico bestial y 2 GB de ram mínimo.

MinSpareServers 100

Pero mejor que tengas mucha ram y una buena cpu. Cada SpareServer será un subproceso del apache, que por lo tanto, consumirá memoria ram.

Número de procesos al arrancar el Apache:

#StartServers 5
StartServers 16

MaxRequestsPerChild 0

Esto es todo, los valores KeepAlive, Timeout, etc, pueden dejarse por defecto. También el MaxClients 150 (por defecto, puedes ponerle un valor más grande). Piensa que serían 150 conexiones simultáneas.

Los valores 0, indican "Ilimitado", es lo recomendado, si es admitido.

También la opción de "HostNameLookups" es mejor ponerla en "HostNameLookups off", ya que así evitamos tener que hacer un reverso de DNS de cada visitante que entra y hace un "hit", y aunque existe una caché de DNS para agilizar este proceso, no es recomendable.

Otra cosa que he leído que dicen que para mejorar el rendimiento, bueno, mejor dicho, para no estropearlo, es usar apache (httpd) vía TCP de wrappers con el inetd.conf. Si necesitas, por ejemplo, bloquear el acceso a alguien usa un .htaccess usando un deny from y no usando inetd.conf. Apache es suficientemente seguro y configurable como para depender de TCP wrappers.

Para probar el rendimiento del apache, puedes hacer un test que lleva incluido el propio apache. Es un ejecutable que se llama "ab" y está en el directorio /bin o /sbin. Este programa simula gran cantidad de tráfico para el Apache y es una muy buena manera de medir el rendimiento de tu apache.

Sino quieres que se muestre la versión del Apache que estás usando:

Busca ServerSignature y lo pones en off:

ServerSignature off

Y añades debajo:

ServerTokens ProductOnly

De esta manera la versión de tu Apache será "Apache" a secas, sin decir la versión exacta (Ej. --> Apache 1.3.29).

ServerTokens permite especificar como opciones

Minimal|ProductOnly|OS|Full.

Configurando Apache versión httpd-2.0.40 como IIS 5.0.

En el fichero: httpd-2.0.40/include/ap_release.h

Cambiamos las líneas:

```
#define AP_SERVER_BASEVENDOR "Apache Software Foundation"
#define AP_SERVER_BASEPRODUCT "Apache"
#define AP_SERVER_BASEREVISION "2.0.40"
#define AP_SERVER_BASEVERSION AP_SERVER_BASEPRODUCT "/"
AP_SERVER_BASEREVISION
#define AP_SERVER_VERSION AP_SERVER_BASEVERSION

#define SERVER_BASEVENDOR "Apache Group"
#define SERVER_BASEPRODUCT "Apache"
#define SERVER_BASEREVISION "1.3.24"
#define SERVER_BASEVERSION SERVER_BASEPRODUCT "/" SERVER_BASEREVISION
```

De manera que queden así:

```
#define AP_SERVER_BASEVENDOR "Microsoft Corp"
#define AP_SERVER_BASEPRODUCT "Microsoft-IIS"
#define AP_SERVER_BASEREVISION "5.0"
#define AP_SERVER_BASEVERSION AP_SERVER_BASEPRODUCT "/"
AP_SERVER_BASEREVISION
#define AP_SERVER_VERSION AP_SERVER_BASEVERSION
```

En el fichero:

httpd-2.0.40/server/core.c

Cambiamos la línea:

```
Static enum server_token_type ap_server_tokens = SrvTk_FULL;
```

De manera que quede así:

```
Static enum server_token_type ap_server_tokens = SrvTk_MIN;
```

En este mismo fichero cambiamos:

```
Static apr_status_t reset_version (void *dummy)
{
version_locked = 0;
ap_server_tokens = SrvTk_FULL;
server_version = NULL;
return APR_SUCCESS;
}
```

Por ésto otro:

```
Static apr_status_t reset_version (void *dummy)
{
version_locked = 0;
ap_server_tokens = SrvTk_MIN;
server_version = NULL;
return APR_SUCCESS;
}
```

Para terminar editamos el fichero:

httpd-2.0.40/os/unix/os.h

Cambiando la línea:

```
#ifndef PLATFORM
#define PLATFORM "Unix"
#endif
```

Para que queden así:

```
#ifndef PLATFORM
#define PLATFORM "Win32"
#endif
```

Problemas comunes

Si en el apache-status está todo el rato "L", o sea "Logging", te está diciendo ni más ni menos, que purges (vacies los logs, pero antes analiarlos con Awstats, WebAlizer, etc), ya deben tener un tamaño consdirable, y al apache le cuesta mucho rato loggear, o ya no tiene espacio.

Si en el apache-status hay muchas conexiones "R", o sea "Reading", significa que las DNS de tu dominio están haciendo el tonto (han caído temporalmente, o algo parecido).

Protegiendo tu ancho de banda con Apache

La mejor manera para proteger el ancho de banda de tu apache es haciendo que nadie pueda linkarte descargas o imágenes. Las técinas de ocultismo en directorios raros o scripts PHP, no son métodos del todo efectivos.

El único requisito previo es tener activado el módulo rewrite.

Vamos a ver un ejemplo haciendo un .htaccess en el directorio que queramos proteger y añadiendo:

RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER ^visitar.php?url=http://yourDomain.dom/.$ [NC]*
RewriteCond %{HTTP_REFERER} !^visitar.php?url=http://www.yourDomain.dom/.$ [NC]*
RewriteRule ..png$ - [L]*
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^visitar.php?url=http://yourDomain.dom/.$ [NC]*
RewriteCond %{HTTP_REFERER} !^visitar.php?url=http://www.yourDomain.dom/.$ [NC]*
RewriteRule ..jpg$ - [L]*

RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !
^visitar.php?url=http://yourDomain.dom/.$ [NC]*
RewriteCond %{HTTP_REFERER} !
^visitar.php?url=http://www.yourDomain.dom/.$ [NC] RewriteRule.*.gif$ - [L]*

Esto permite que las imágenes png, jpg y gif de ese directorio puedan ser vistas por nosotros, pero no linkadas o vistas desde otros dominios.

Protegiendo que te linkeen (segunda parte)

RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^visitar.php?url=http://(www.)?site.com/.*$ [NC]
RewriteRule. (gif|jpg)$ - [F]

RewriteRule. (gif|jpg)$
visitar.php?url=http://www.site.com/stop.gif [R, L]

Falta cambiar "site.com" y stop.gif with por tus nombres reales. Al intentar ver una imagen desde otro dominio aparece la imagen stop.gif.

¿Verdad que es fácil? Modo mod_write activado y expresiones regulares y marchando.

Hay otras maneras, si tienes un descargas en archivos "zip", puedes proteger este archivos mediante contraseña. También puedes incluir un readme o un .url dentro del zip.

Otra buena manera de proteger tu ancho de banda (especialmente en las descargas) es utilizando la "Autorización del Apache" que se ve en el punto 8.

Veamos ahora un ejemplo:

Queremos proteger el directorio descargas:

<Directory "/home/pepe/public_html/descargas">
AuthUserFile /home/pepe/htpasswd
AuthType Basic
AuthName "Escribe un nombre de usuario y contraseña"

require valid-user
</Directory>

De esta manera para descargar cualquier archivo o documento que haya en la carpeta descargas será necesario introducir un nombre de usuario y contraseña válida. De la misma manera si hay una carpeta dentro de descargas (descargas/otras descargas) quedará también automáticamente protegida.

Para aprender a crear el htpasswd consulta el punto 8.

Otra manera pasa por subir las descargas a otro lugar y sin borrarlas de tú maquina incluir este sencillo código:

Redirect permanent /descargas
visitar.php?url=http://www.miotraweb.com/descargas

Todas las descargas se bajarán de "miotraweb.com", aunque estén en nuestra máquina.

Autorización del Apache basada en host (ip):

<Directory /privado>
order deny,allow
deny from all
allow from ip1
allow from ip2
</Directory>

Autorización del Apache basada en usuario:

Primero crear el usuario con el binario (el binario, el .exe en Windows, se encuentra por defecto en la carpeta bin del apache y en Linux en /usr/local/apache/bin):

htpasswd -c htuser alex

Te pedirá la contraseña de alex dos veces.

Nos crea el archivo htuser que es donde está la contraseña de alex y tenemos que llamar a ese archivo en el httpd.conf

Ejemplo en Windows:

AuthUserFile "C:\Archivos de programa\Apache Group\Apache2\bin\htuser"

Ejemplos en Linux:

#el password está en el archivo htpasswd
AuthUserFile /home/alex/htpasswd
AuthUserFile conf/htuser
AuthGroupFile conf/htgroup

El archivo htuser debe estar en conf, y sino le dices la ruta :

Lo mejor es poner este archivo de password fuera del httdocs que tengamos, ya que aunque la contraseña está encriptada, es mejor no jugársela y que algún listillo nos cracker el pass.

Quedaría así:

<Directory /privado>
AllowOverride None
Options Index
#donde está el fichero con lo passwords
AuthUserFIle conf/htuser
AuthGroupFile conf/htgroup
#nombre del recurso
AuthName "Privado"
#tipo
AuthType Basic
require user alex pepe
requiere group admin.
 # o bien
require valid-user
</Directory>

3.2 Internet Information Server

IIS, es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del *Option Pack* para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a un computador en un servidor de Internet o Intranet es decir que la computadora que tienen este servicio instalado se puede publicar páginas web tanto local como remotamente (servidor web). Si se quiere para usar local es mas recomendable utilizar el PWS (Personal Web Service) El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

3.2.1 Arquitectura de IIS 6.0

Los Servicios Internet Información versión 6.0 corre en todas las ediciones del sistema operativo Microsoft® Windows® Servidor 2003 esta nueva arquitectura ofrece flexibilidad por tener la opción de dos modos de aislamiento de aplicación. La nueva arquitectura le ayuda a ejecutar un servicio Web más rápido, fiable y seguro.
IIS 6.0 proporciona un servicio Web rediseñado que publicado en el (WWW) arquitectura que puede ayudarlo a lograr fiabilidad, escalabilidad, y seguridad para sus sitios Web, si ellos corren en un solo servidor IIS corriente o en los servidores múltiples.

El aislamiento de la aplicación es la separación de aplicaciones por límites del proceso que impiden a una aplicación o un sitio Web afectar a otro y reducir el tiempo que usted gasta reiniciando los servicios para corregir los problemas relacionados a las aplicaciones.

En IIS 6.0, se configura el aislamiento de la aplicación diferentemente para cada uno de los dos modos de aislamiento de la aplicación. Ambos modos confían en el HTTP de la pila protocolar (también llamado HTTP.sys) esto se lo hace para recibir las demandas de la Internet y contestaciones del retorno. HTTP.sys reside en el kernel dónde se encuentra el código del sistema operativo, HTTP.sys escucha las demandas de HTTP.

La nueva arquitectura de IIS 6.0 y el ambiente de aislamiento de aplicación permiten a aplicaciones de Web individuales que siempre corren en el modo del usuario funcionar dentro de un proceso del obrero autónomo. Un proceso del obrero es código modo usuario cuyo papel es procesar las demandas, como devolver una página estática o invocar un Internet Servidor API (ISAPI) extensión o filtro. Los procesos del trabajador usan HTTP.sys para recibir las demandas y enviar las contestaciones encima de HTTP.

Como se ha podido observar las principales partes de la arquitectura de IIS 6.0 son:

- La metabase
- El http.sys
- Procesos del trabajador
- El W3SVC
- Modo de aislamiento

La metabase de IIS 6.0

La metabase es una tienda jerárquica de información de la configuración y esquema que se usan para configurar los Servicios de Internet Información (IIS) 6.0.

La metabase es un almacenamiento jerárquico de los valores de la configuración usados por IIS, que tienen incorporados una amplia funcionalidad, como por ejemplo, herencia, tipos de datos y, por supuesto, seguridad .Esto se puede ver editando el archivo %systemroot%\system32\inetsrv\MetaBase.xml. Cuando se edita la metabase es posible agregar, borrar o cambiar en tiempo real los Application Pools, entre otras cosas, mediante el uso de scripts o código estándar y cambiando el comportamiento del IIS en forma dinámica. Increíble, ¿no? Sí, pero real.

El archivo de configuración de la metabase, es MetaBase.xml, este guarda la mayoría de la configuración de IIS. Algunos valores de configuración de IIS también se guardan en el registro de Windows. Si un valor de la configuración es uno que usted podría necesitar configurar o cambiar, o si usted puede acceder al archivo en el IIS 6.0 por medio de una interfaz del usuario, entonces el archivo se guarda típicamente en la metabase de IIS.

El archivo MetaBase.xml es organizado en una estructura jerárquica que puede variar dependiendo de las opciones que son hechas cuando IIS se instala y, si se necesita reconfigurarlo. Cuando IIS se empieza o se reinicia, las escenas de la configuración se leen del archivo MetaBase.xml y se copian en memoria en el IIS que está llamado en la metabase. Cuando se usa el administrador de IIS o las interfaces programáticas para cambiar la configuración de IIS, los cambios se hacen en la memoria de la metabase y se graban en el archivo MetaBase.xml después de un número previamente configurado de cambios, o según los intervalos de tiempo regulares.

En las condiciones fisicas, la metabase está una combinación de los archivos MetaBase.xml y MBSchema.xml y la memoria de la metabase. La información de configuración del IIS se guarda en el archivo de MetaBase.xml, mientras que el esquema de la metabase se guarda en el archivo MBSchema.xml.

Cuando IIS empieza, estos archivos se leen por la capa del almacenamiento y entonces escritos en la memoria de la metabase a través de los Administradores de Base de Objetos (ABOs), como se muestra en la figura.

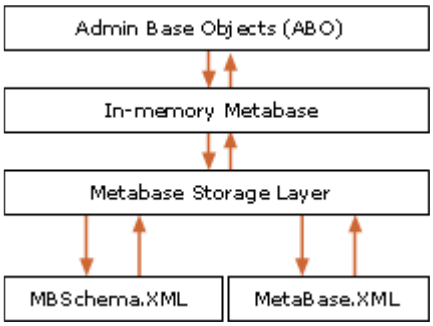

Figura 11 Capas de Comunicación de la metabase

Se guardan la configuración de la metabase y el esquema en los nodos separados en la memoria de la metabase. La propia memoria de la metabase reside en el archivo del IIS archivo que se encuentra en la RAM de su computadora. Mientras IIS está corriendo, se ahorran cambios que se hacen a la memoria de la metabase periódicamente en el disco duro. La memoria de la metabase también se detiene su información al disco duro cuando IIS se detiene. El archivo que aloja a la metabase es el archivo Inetinfo.exe. Este archivo además de alojar a la metabase; también aloja los servicios de FTP/SMTP/NNTP.

El http.sys

IIS 6.0 usa HTTP.sys que es parte del subsistema de la gestión de redes del sistema operativo de Windows.

El http.sys reemplaza el uso de Winsock para IIS que es un componente del modo-usuario para recibir las demandas de HTTP.

Usando HTTP.sys y los nuevos servicios de WWW para procesar las demandas, IIS 6.0 entrega las mejoras siguientes:

- El modo-kernel escondido. Sirve para peticionar las contestaciones escondidas sin cambiar al modo del usuario.
- El modo-kernel petición de cola. Demanda formación de colas de espera. Las demandas causan menos cambios al contexto, porque el kernel pide directamente al proceso del obrero correcto. Si ningún proceso del obrero está disponible aceptar una demanda, el modo-kernel petición de cola sostiene la demanda hasta que un proceso del obrero lo recoja.

HTTP.sys en la arquitectura de IIS 6.0 proporciona los beneficios siguientes:

- Cuando hace falta un proceso de obrero, el servicio no se interrumpe; el fallo es indetectable por el usuario porque el kernel hace cola de peticiones mientras los WWW reparan las salidas de un nuevo proceso del obrero para esa aplicación pool.
- Las peticiones son procesadas más rápidamente porque estas se dirigen directamente del kernel al proceso de obrero del modo-usuario apropiado; en lugar de dirigirse a dos procesos del modo-usuario.

Cuando usted crea un sitio Web, IIS registra el sitio con HTTP.sys que entonces recibe cualquier demanda de HTTP para el sitio. HTTP.sys funciona como un expedidor, mientras envía el Web recibe a la cola de la petición para el proceso del modo-usuario que ejecuta el sitio de Web o aplicación de Web. HTTP.sys también envía atrás las contestaciones al cliente. De otra manera recupera una contestación guardada de su escondite interior, HTTP.sys no procesa las demandas que recibe. Por consiguiente, ningún código específico está esta cargado en el modo del kernel. Como resultado, los virus en el código de una aplicación específica no pueden afectar el kernel o pueden llevar al fracaso al sistema.

HTTP.sys proporciona los siguientes servicios en IIS 6.0:

- Encamina las demandas de HTTP a la cola de la demanda correcta.
- Esconde las contestaciones en el modo-kernel.
- Ejecuta todo lo basado en el texto para el servicio de WWW.
- Implementa la Calidad de Servicio (QoS) funcionalidad que incluye la conexión limitada, interrupciones de conexión, la longitud de cola limitada, y control de ancho de banda.

Cuando IIS 6.0 corre en el proceso del trabajador en modo aislado, HTTP.sys escucha las demandas y hacen cola esas demandas en la cola apropiada. Cada cola de la demanda corresponde a una aplicación pool. Una aplicación pool corresponde a una cola de la demanda dentro de HTTP.sys y uno o más procesos del trabajador.
Si una aplicación causa el proceso de trabajador de modo-usuario para terminar inesperadamente, HTTP.sys continúa aceptando y hace cola las peticiones, con tal de que el servicio WWW todavía está corriendo, las colas todavía están disponibles.

Cuando el servicio WWW identifica un proceso de trabajador enfermo, empieza un nuevo proceso de trabajador si las demandas excelentes están esperando ser reparadas. Así, aunque una ruptura temporal ocurre en el proceso repetición de

modo-usuario, un usuario final no experimenta el fracaso porque se mantienen las conexiones de TCP/IP, y las demandas continúan siendo hechas cola y procesadas.

En resumen el http.sys realiza las siguientes funciones:

- Controlador de dispositivos en modo kernel
- Recibe solicitudes HTTP
    - Enruta las solicitudes hacia los procesos del trabajador
    - Envía respuestas HTTP
- *No* procesa las solicitudes
- Realiza otros servicios

Procesos del trabajador

Un proceso del trabajador es código del modo-usuario cuyo papel es procesar las demandas, como por ejemplo devolver una página estática, mientras se esta invocando una extensión de ISAPI, se filtra, o se ejecuta una Interfaz de la Entrada Común (CGI) negociante, además este procesa las solicitudes Web.

En ambos modos de aislamiento de aplicación, el proceso del trabajador se controla por el servicio WWW. Sin embargo, en el proceso del trabajador modo aislado, un proceso del trabajador corre como un archivo ejecutable nombrado W3wp.exe, y en IIS 5.0 modo de aislamiento, un proceso del trabajador se organiza por Inetinfo.exe. en la arquitectura para IIS 5.0 modo de aislamiento se usa una línea punteada para hacer pensar en la relación entre el proceso del trabajador y el servicio de WWW.

El proceso del trabajador usa HTTP.sys para recibir las demandas y enviar las contestaciones usando HTTP. Los procesos del trabajador también son ejecutados en el código de la aplicación, como las aplicaciones de ASP.NET y XML. Se puede configurar IIS para ejecutar múltiples procesos del trabajador que sirven para que diferentes aplicaciones se agrupen concurrentemente. Este plan separa las aplicaciones por los límites del proceso y logra que el servidor Web tenga una fiabilidad máxima.

Por defecto, el proceso del trabajador en modo aislado corre bajo la cuenta de Servicio de Red que tiene la seguridad más fuerte (el menor acceso) compatible con la funcionalidad que se requiere.

El proceso del trabajador ejecuta aplicaciones en modo usuario, procesa archivos w3wp.exe nombrados, el rol principal que tiene este proceso es el de procesar las solicitudes, además de devolver páginas estáticas, invocar extensiones ISAPI, ejecutar manejadores CGI y ejecuta código de aplicación. También utiliza al http.sys para enviar/recibir y este proceso es administrado por el W3SVC.

Servicio de administración Web (W3SVC)

Este servicio esta encargado de administrar los procesos del trabajador. En el momento de inicialización:

- Construye la tabla de enrutamiento del espacio de nombre http.sys
- Inicia los procesos del trabajador
    - Cuando se recibe la primera solicitud en http.sys
- Administra los procesos del trabajador
- Supervisa la salud del proceso del trabajador
- Inicia/detiene, recicla, entre otros
- No procesa las solicitudes

Modos de aislamiento de IIS

IIS 6.0 soporta dos modos de aislamiento de procesos

- El modo de aislamiento del trabajador. El cual es el modo preferido para IIS 6.0
- Modo de aislamiento de IIS 5.0

Este modo de aislamiento proporciona una compatibilidad hacia atrás para aplicaciones Web, es similar a IIS 5.0, donde cada solicitud debe cruzar Inetinfo.exe, este modo de aislamiento no tiene grupos de aplicaciones ni reciclado

Valores predeterminados en el modo de aislamiento

Los valores predeterminados son diferentes dependiendo de la instalación

- Nueva instalación: Proceso del trabajador
- Actualización de IIS 4.0 o 5.0: Modo IIS 5.0
- Actualización de IIS 6.0: Se mantiene el modo

Recomendaciones de actualización:

- Configurar un nuevo PC que ejecute IIS 6.0 en el modo de aislamiento del proceso del trabajador
- Probar las aplicaciones en nuevo PC con IIS 6.0
- Migrar las aplicaciones a un nuevo servidor después de terminar la prueba

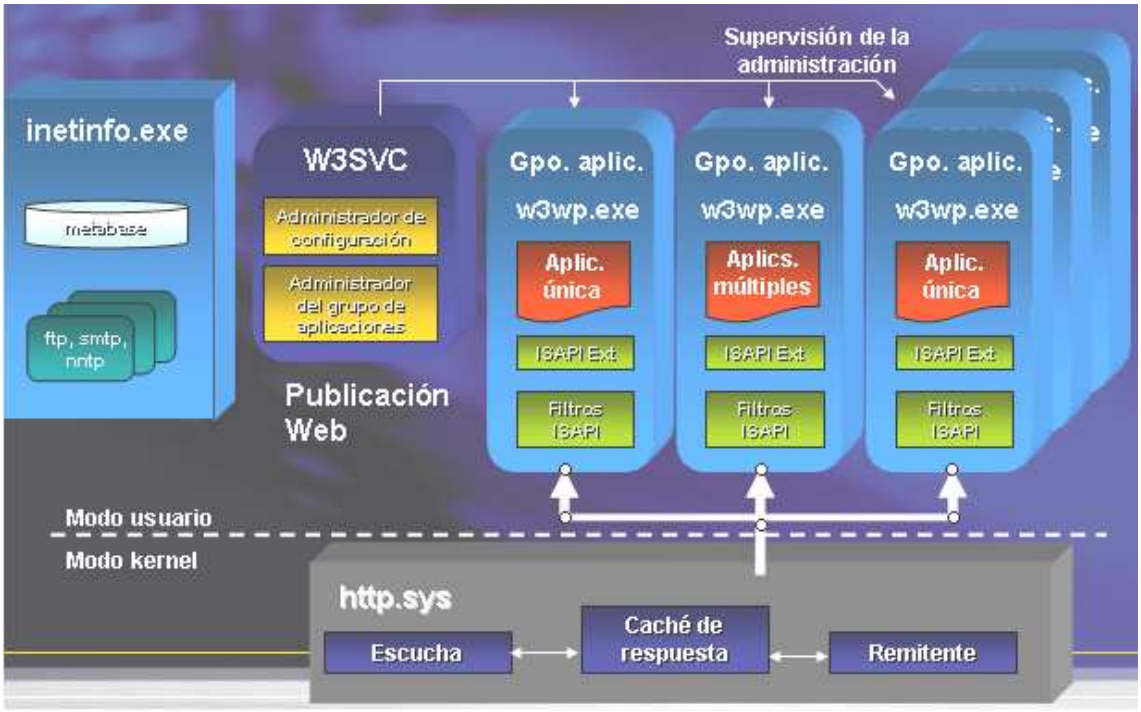

Figura 12 Arquitectura de IIS 6.0

Luego de haber examinado la arquitectura que compone a IIS 6.0 se mostrará algunas de las mejoras que incluye este servidor Web en relación con sus antecesores (Internet Information Server 4.0, Internet Information Server 5.0, Internet Information Server 5.1):

Aislamiento de WWW Service Administration and Monitoring.

Uno de los mayores cambios que afectan directamente a la fiabilidad tienen que ver con la manera en que IIS 6.0 funciona por dentro. Para que sirva de comparación, echemos un rápido vistazo a la arquitectura de IIS 5.0. En IIS 5.0, el proceso fundamental que ejecuta las funciones básicas de IIS es inetinfo.exe. Inetinfo también alberga aplicaciones web que usted establece para ejecutarse en protección baja de aplicación (en proceso) en la consola *Internet Information Services* de *Microsoft Management Console* (MMC). También puede ejecutar aplicaciones web "fuera de proceso" fijando la protección de aplicación a media (agrupada) o alta (aislada), en cuyo caso se ejecutan en un proceso denominado dllhost.exe. Inetinfo tiene otras obligaciones relevantes, como enrutar las peticiones entrantes desde Winsock a las aplicaciones web adecuadas fuera de proceso, y albergar IIS Admin Service, que se comunica con la metabase (es decir, el almacén de configuración de los valores para los sitios y directorios web de IIS). Como resultado, las tareas relacionadas con la configuración y la administración interna del servidor web se ejecutan en el mismo

proceso como aplicaciones web. Por tanto, una aplicación mal escrita puede arrojar una excepción (por ejemplo, una caída) o bien interrumpir Inetinfo y parar todo el servidor web, incluidas las aplicaciones fuera de proceso.

En IIS 6.0, el componente *WWW Service Administration and Monitoring* se ejecuta en un proceso independiente de las aplicaciones web, de manera que una aplicación caída o "colgada" no puede producir el fallo de *WWW Service Administration and Monitoring*. Esta independencia permite que *WWW Service Administration and Monitoring* controle el servidor Web y lo mantenga en funcionamiento; *WWW Service Administration and Monitoring* puede incluso reiniciar las aplicaciones que fallan para responder a consultas del tipo "¿estás vivo?" desde el componente.

Pools de aplicaciones

Las aplicaciones web de IIS 6.0 se ejecutan en un proceso de trabajo denominado w3wp.exe, el sustituto de Dllhost de IIS 5.0 (y de wam.exe de IIS 4.0). Sin embargo, en IIS 6.0 puede dar un nombre a un proceso de usuario y gestionarlo como un elemento que se denomina *pool* de aplicaciones. Puede asignar fácilmente un sitio web, directorio o directorio virtual a cualquier pool de aplicaciones utilizando el cuadro de diálogo *Properties* de ese elemento, tal y como muestra la **Figura 15**.

Figura 13 Asignando a un sitio Web un pool de aplicaciones

Tenga en cuenta que cada *pool* de aplicaciones es independiente de los demás, de forma que si un *pool* falla o se recicla, sólo se verán afectadas las aplicaciones que se encuentren en dicho *pool*, mientras que los otros grupos de aplicación seguirán

distribuyendo contenido. Esta funcionalidad es quizás el elemento central de las mejoras de fiabilidad de IIS 6.0. Puede organizar sus aplicaciones en los *pools* que mejor se ajusten a sus necesidades. Por ejemplo, puede establecer un *pool* aplicaciones que se ejecuten de manera fiable en un *pool*, y aislar aplicaciones que no resulten fiables o fidedignas en grupos independientes para aislar las aplicaciones fiables del "comportamiento díscolo" de las aplicaciones no fidedignas.

WWW Service Administration and Monitoring comprueba automáticamente que un *pool* de aplicaciones responde cada 30 segundos (por defecto) o en un intervalo que pueda definir en la pestaña *Health* del cuadro de diálogo *Properties* del *pool* de aplicaciones. Si un grupo no responde, *WWW Service Administration and Monitoring* recicla o reinicia dicho grupo sin la intervención de un administrador, lo cual supone una mejora significativa de la fiabilidad. Además, puede utilizar los valores de configuración de la pestaña *Recycling*, que muestra la **Figura 16** para configurar el momento en que dicho grupo reciclará (es decir, parará e iniciará). Por ejemplo, los grupos de aplicaciones se reciclan por defecto cada 1740 minutos (es decir, 29 horas). Si ese valor no se ajusta sus necesidades puede establecer que el reciclaje se produzca a una hora concreta del día, por ejemplo, a las 3:00H hora local. El reciclaje programado puede resultar útil para las aplicaciones que necesitan restablecerse periódicamente.

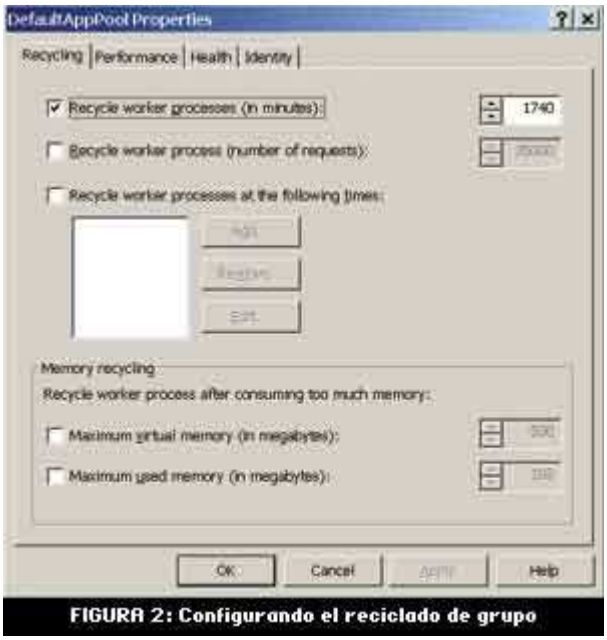

Figura 14 Configurando el reciclaje de Grupo

La pestaña *Recycling* también le permite supervisar el consumo de memoria de un proceso de usuario y reciclar la aplicación cuando exceda un límite determinado. La

capacidad de reciclar automáticamente aplicaciones que consuman recursos en exceso puede ayudar a numerosos servidores IIS que deban distribuir aplicaciones que con el tiempo consuman demasiada memoria de manera inexplicable. Para más información acerca de los procesos de reciclaje de IIS 6.0, consulte el apartado “Cómo funciona el reciclaje”.

CÓMO FUNCIONA EL RECICLAJE

IIS 6.0 utiliza una funcionalidad denominada reciclaje superpuesto (*overlapping recycling*) para reciclar los procesos de usuario. De forma resumida, el proceso de reciclaje funciona de la siguiente manera:
Un proceso de usuario se marca para reciclarse mediante *WWW Service Administration and Monitoring* o es solicitado por la aplicación.

IIS 6.0 crea otro proceso de usuario para sustituir el proceso marcado. En este punto, ambos procesos de usuario se encuentran en memoria, y quizás activos en al mismo tiempo.

Se envían nuevas peticiones al nuevo proceso de usuario, mientras que el proceso marcado sigue sirviendo peticiones en su cola, si es que puede.

Cuando la cola del proceso de usuario marcado se encuentra vacía o se produce un tiempo de espera (en el caso de un proceso de usuario que no responda), IIS 6.0 destruye el proceso de usuario a menos que el valor OrphanWorkerProcess de la metabase esté establecido. En ese caso, IIS 6.0 mandrá el proceso fallido para su depuración.

El valor OrphanWorkerProcess ha demostrado ser muy valioso en la depuración de procesos de usuario fallidos. Para más información acerca de este valor, consulte los archivos de ayuda de IIS 6.0.

Figura 15 Como funciona el reciclaje

Cuando configura el reciclaje de sus grupos de aplicaciones, probablemente querrá que IIS 6.0 registre los eventos de reciclaje en el Visor de eventos en el momento en que se produzcan. Por defecto, sólo se registrarán aquellos eventos de reciclaje relativos a la memoria y al tiempo. Deberá mirar en los archivos de Ayuda de IIS 6.0 para encontrar la propiedad de la metabase *LogEventOnRecycle*. Esta propiedad es un valor de máscara de bits en la que cada bit activa o desactiva el registro de un evento específico. Puede utilizar el siguiente comando para activar el registro de todos los eventos:

```
cscript SystemDrive\Inetpub\
AdminScripts\adsutil.vbs
Set W3SVC/AppPooLs/
LogEventOnRecycle
```

Aplicaciones con monitorización automática

La capacidad de configurar grupos de aplicaciones de manera que se reciclen cuando se produzcan ciertos eventos resulta útil, pero más aún lo es la capacidad de crear chequeos del estado justo en una aplicación y, si es necesario, que dicha aplicación solicite el reciclaje. Cualquier extensión API de IIS 6.0 Internet Server (ISAPI), incluidas aquellas que usted cree, puede utilizar la nueva función HSE_REQ_REPORT_UNHEALTHY para invocar el reciclaje de la aplicación. Esta

capacidad le permite realizar chequeos de estado directamente en sus extensiones ISAPI, en lugar de realizarlas exclusivamente en el servidor.

Active Server Pages (ASP) y ASP.NET en IIS 6.0 cuentan con funcionalidades de monitorización automática, y bajo ciertas circunstancias, invocarán a HSE_REQ_REPORT UNHEALTHY.ASP, por ejemplo, controla el tiempo que tarda el motor del script ASP para regresar desde una petición de procesamiento. Si el tiempo excede el valor de configuración de *ASP Script Timeout* en la consola *Internet Information Services*, IIS ordenará al motor del *script* que pare el procesamiento. Si dicho motor no consigue regresar de la petición de parar, ApSP abandonará el proceso. ASP realiza un seguimiento interno del número de procesos abandonados. Si dicho número es suficiente, asp.dll solicitará el reciclaje.

Afinidad con la CPU

Los grupos de aplicaciones cuentan con otra funcionalidad que puede resultar útil en sistemas con varias CPUs. Se denomina afinidad con la CPU. Dicha funcionalidad le permite designar cuáles serán las CPU que utilice para sus aplicaciones web, y puede resultar útil si está obligado contractualmente a designar una CPU para una aplicación concreta del cliente. Para establecer la afinidad entre un *pool* de aplicaciones y una CPU, debe fijar dos propiedades de metabase: *SMPAffinitized* y *SMPProcessorAffinityMask*.

Debe utilizar la propiedad *SMPAffinitized* para activar la afinidad para un *pool* de aplicaciones. En *Adsutil*, puede utilizar el siguiente comando para establecer esta propiedad:

```
Cscript SystemDrive\Inetpub\
AdminScripts\adsutil.vbs
Set W3sVc/AppPooLs/
ApplicationPoolName/
SMPAffinitized TRUE
```

Donde *ApplicationPoolName* es el nombre del grupo con el que desea establecer la afinidad. Cuando *SMPAffinitized* se establece en el valor por defecto de 0 (*FALSE*, utilizando *Adsutil*), el *pool* de aplicaciones podrá ejecutarse en cualquier procesador.
La propiedad *SMPProcessorAffinityMask* es una máscara de bits facilitada como número hexadecimal que representa a la CPU que desea asignar al grupo. También puede asignar un grupo para que se ejecute en más de un procesador, si su sistema

tiene varios procesadores. El siguiente comando configura un grupo para ejecutarse en los procesadores 0 y 2:

```
Cscript SystemDriveVInetpub\
AdminScripts/adsutil.vbs
SetW3SVC/AppPools/
ApplicationPoolName/
SMPProcessorAffinityMask 0x5
```

El Kit de recursos de *Internet Information Services* (IIS) 6.0 cuenta con una tabla de valores hexadecimales de procesador.

Granjas Web

IIS 6.0 puede realizar también otro truco en relación con los *pools* de aplicaciones: su funcionalidad de granja web le permite establecer que un *pool* de aplicaciones contendrá más de un proceso de usuario. Cuando se configura de esta manera, IIS 6.0 creará una instancia de w3wp.exe para cada petición, hasta llegar al número de procesos de usuario que especificó para el *pool*. Por ejemplo, si establece que un *pool* de aplicaciones tenga tres procesos de usuarios después de tres peticiones, verá que se ejecutan tres instancias de W3wp en su servidor, todas ellas distribuyendo el mismo contenido. IIS 6.0 realiza conexiones subsiguientes a los procesos de usuarios en el jardín web a modo de ronda. Aunque las granjas web no tienen una afinidad de sesión per se, IIS6.0 canaliza todo el tráfico desde una conexión al mismo proceso. El proceso de usuario guarda la información de la sesión, la clave secreta de la Capa de Conexión Segura (*Secure Socket Layer*, SSL) y la información de autentificación hasta que se finalice la conexión o se recicle el proceso de usuario.

Las granjas web no se utilizan de manera general. En nuestra opinión, esto se debe en parte porque el rendimiento y la fiabilidad de IIS 6.0 son, por lo general, buenos o incluso excelentes sin ellos, y también porque la gente no sabe qué tipo de aplicaciones pueden beneficiarse de su uso. No resulta sencillo encontrar ejemplos de instalaciones reales en las que esta funcionalidad beneficie claramente a la aplicación. Si conoce alguna, sería interesante que nos lo comunicara. No obstante, a continuación ofrecemos un par de ejemplos.

Una granja web podría resultar útil cuando su aplicación utilice un componente que tenga una restricción incorporada. Por ejemplo, si tiene un objeto de conexión a base

de datos que sólo acepte 10 conexiones, podrá habilitar n procesos de usuario para facilitar 10 x n conexiones.

Otro escenario más probable es una conexión de base de datos que "bloquee" una aplicación. En este caso, la aplicación realiza una petición a la base de datos que tarda cierto tiempo en ejecutar. El proceso, (es decir, la unidad de trabajo para ejecutar el código en la CPU) que realiza la petición está ahora paralizada esperando en dicha petición, y no se encuentra disponible para otra tarea. Por lo tanto, el *pool* disponible de procesos de usuario se reducirá a 1. Si las peticiones se realizan antes de que la base de datos pueda procesarlas, su aplicación podría quedarse "excluida del proceso". El escenario se beneficiaría de la granja web porque cada proceso en dicha granja tiene su propio *pool* de procesos dedicado.

Tenga en cuenta que ASP.NET también incorpora funcionalidades de granja web y afinidad de procesador, que se diferencian totalmente de la implementación de IIS 6.0. Puede encontrar los valores de configuración para dichas funcionalidades en el elemento *processModel* del archivo XML machine.config: el principal archivo de configuración para las aplicaciones ASP.NET. IIS 6.0 ignora totalmente los valores de machine.config para el jardín web y la afinidad de procesador; sólo IIS 5.x los utiliza.

Windows System Resource Manager (WSRM)

Puede perder todas esas ventajas de aislamiento de procesos que ofrecen los pools de aplicaciones si sus aplicaciones consumen recursos de sistema de manera excesiva. Por supuesto, podrá configurar una aplicación para que se recicle cuando se utiliza demasiada memoria, pero ¿no preferiría poder asignar el máximo espacio de memoria RAM para que lo utilice la aplicación? Además, si su aplicación consume muchos recursos de la CPU, IIS 6.0 no cuenta con demasiadas funcionalidades incorporadas para ayudarle, porque su control incorporado de la CPU hace poco más que habilitar y deshabilitar la aplicación. Como aplicación en modo usuario, IIS 6.0 simplemente no tiene demasiados derechos para gestionar lo que se produce en modo *kernel*, y, por lo tanto, no puede autoregular esas actividades.

Para resolver esos problemas, Microsoft ofrece un nuevo servicio importante denominado *Windows System Resource Manager*. WSRM se ejecuta exclusivamente en Windows 2003, Enterprise Edition y Windows 2003, Datacenter Edition, pero puede gestionarlo desde cualquier servidor Windows 2003. Con WSRM instalado puede

configurar un proceso (incluidos los servicios) de forma que no utilizará más de un volumen concreto de memoria o de capacidad de la CPU. Además, puede asignar aplicaciones a CPU específicas. Por consiguiente, podrá designar aplicaciones que deban tener una alta disponibilidad a CPU específicas, y asignar otros procesos y servicios para ejecutarse en la CPU restante; sin utilizar la de afinidad de procesador de IIS. El control de rendimiento, la capacidad de imponer límites basados en la fecha y el tiempo, la capacidad de incluir y excluir usuarios y las funciones de registro también están incorporadas en WSRM. Si necesita asegurar memoria o el uso de CPU de forma fiable en su servidor, WSRM es la solución.

BITS

Una nueva funcionalidad, que prácticamente ha pasado inadvertida, y a la que nosotros llamamos "servicio de goteo", es la denominada *Background Intelligent Transfer Service* (BITS). Aparte de lo divertidas que resultan las caras de extrañeza de sus compañeros cuando le oyen nombrar al "servicio de goteo", BITS tiene el objetivo práctico de conservar el ancho de banda, y, por lo tanto, incrementar la disponibilidad en los casos de cargas importantes.

Cuando se encuentra instalado, BITS gestiona descargas y cargas desde y hacia el servidor utilizando el "ancho de banda sobrante" para distribuir contenido al cliente. BITS también realiza un seguimiento de las descargas solicitadas y le permite realizar colas de peticiones y resúmenes de las mismas. Tal y como muestra la **Figura 18**, puede configurar varios parámetros, incluido el tamaño máximo de los archivos, la frecuencia de depurar la cola de trabajos no realizados y la posibilidad de habilitar soporte limitado para las granjas de servidores. Si alguna vez ha descargado archivos de gran volumen desde MSDN, reconocerá la ventana del lado cliente que muestra el estado de su actividad de descarga.

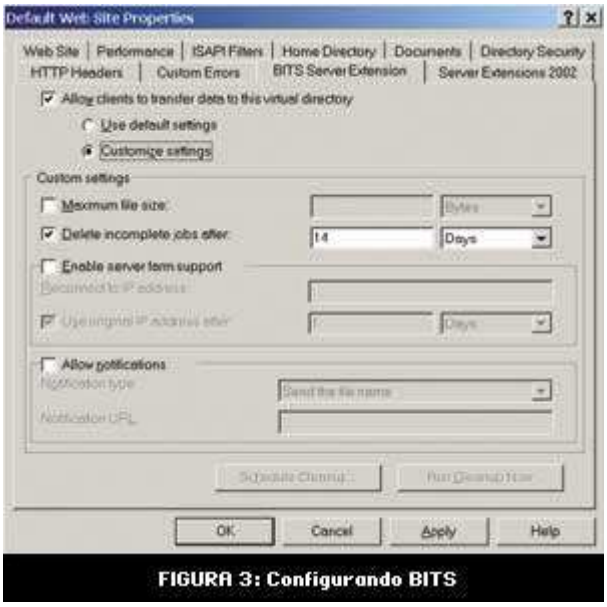

Figura 16 Configurando BITS

El servicio BITS no se instala por defecto cuando usted instala IIS. Podrá hacerlo en los servidores Windows 2003 abriendo el Panel de Control y seleccionando Agregar/Eliminar Programas, Componentes Windows, Servidor de Aplicaciones, *Internet Information Services* (IIS), *Background Intellingent Transfer Service Server Extensions*.

Asegúrese de consultar los archivos de Ayuda para obtener detalles importantes acerca de la seguridad y los directorios virtuales que están habilitados para BITS. BITS ignora los valores de configuración de seguridad de IIS 6.0 y le permite cargar archivos a una carpeta que tenga el permiso de escritura basado en IIS deshabilitado. Se exigen los permisos normales de seguridad de NTFS. A modo de complemento, BITS cuenta con un kit de desarrollo de *software* (SDK). BITS funciona con SSL, es totalmente programable y utiliza autentificación Windows y NTFS, así que quizás sea ese sustituto tan necesario de FTP para publicar archivos en un servidor IIS sin utilizar *Microsoft FrontPage Server Extensions*.

Fiabilidad mejorada

Como hemos dicho antes, IIS 6.0 incluye lo siguiente:

- Aplicaciones aisladas que puede aislar desde aplicaciones web que presenten un comportamiento erróneo.
- Reciclaje automático de aplicaciones que no respondan mediante un servicio independiente.

- Reciclaje automático o procesos de usuarios basados en eventos programables o parámetros configurables.
- Asignación de pools de aplicaciones para CPU designadas.
- Consumo limitado de memoria y tiempo de CPU por parte de la aplicación.
- Cargas y descargas de archivos gestionados que no consumen ancho de banda en exceso.

Utilitarios por línea de comandos

Otras de las mejoras de IIS es la inclusión de varios utilitarios por línea de comandos, entre los cuales podemos citar:

- **iisweb**: permite crear, borrar, arrancar, parar y pausar sitios web, ej: c:\> iisweb /create c:\webroot "mi sitio" /b 192.168.0.2
- **iisback**: permite resguardar y restaurar entre otras cosa sitios webs.
- **iisftp**: permite crear, borrar, arrancar, parar y pausar sitios ftp, ej: c:\> iisftp /create c:\inetpub\ftproot "mi ftp" /b 21 Y también

podemos agregar: iisapp, que permite reportar los identificadores de procesos, iiscnfg, que permite importar y exportar partes seleccionadas del servidor IIS, iisvdir, que permite crear directorios virtuales, e iisftpdr que permite crear directorios virtuales en los sitios FTPs.

Autentificación de usuarios

Como características adicionales de IIS, no podemos dejar de destacar la autentificación de los usuarios que usan las credenciales Passport. Esta integración permite a los administradores tomar ventaja de una base de más de 150 millones de cuentas, sin tener que preocuparse por administrarlas en lo más mínimo. Los pasaportes pueden ser asociados a las cuentas del usuario y una vez que la autenticidad del pasaporte ha sido verificada con la asociación que fue creada previamente, se le crea al usuario un testigo de acceso que le permite presentarse al subsistema de seguridad para intentar acceder a los recursos.

En la actualidad ya existen varios sitios usando este tipo de tecnología y entre ellos podemos mencionar a hotmail.com, betaplace.com, windowsbeta.com, etc.

Servicios FTP

Con relación a los servicios de FTP, una de las nuevas características es el aislamiento de usuarios que permite a los ISPs ofrecer a sus clientes directorios individuales para

la carga de archivos y contenido Web. El aislamiento de usuarios evita que los mismos vean o modifiquen de alguna manera el contenido de otros directorios, sencillamente porque no tienen siquiera visibilidad de los mismos.

Los usuarios no pueden navegar "hacia arriba" de sus propios directorios puesto que estos simulan ser directorios raíz, aún cuando realmente no los son.

Estas funcionalidades y otras (como la Calidad de servicio -QoS-, políticas para restringir el ancho de banda para IIS) permiten afirmar que ISS 6.0 es mucho más fiable que cualquier otra versión anterior del servidor IIS. Añada las mejoras de rendimiento y seguridad de IIS 6.0 a los excelentes comentarios que se ha escuchado de aquellos que han implantado IIS 6.0 y tendrá más de una buena razón para pasarse a la última versión del servidor web.

3.2.2 Directivas de Configuración

Para tener un conocimiento más a fondo de lo que comprenden las directivas de configuración, se debe dar un vistazo a los que son los registros de IIS.

Registros del Internet Information Server 6.0

Es posible recopilar información acerca de la actividad de los usuarios si habilitamos este registro para los sitios Web. La información se almacena en archivos ASCII. Este registro tiene muchas posibilidades y supera pon mucho el ámbito de las características del conocido registro de sucesos de Windows. Los registros pueden incluir información referente a quién ha visitado el sitio, qué ha visto el visitante y cuándo se vio la información por última vez. Podemos utilizar los registros para evaluar la popularidad del contenido o identificar los cuellos de botella de la información.

Podemos configurar los sitios Web o FTP para que graben estas entradas de registro generadas por la actividad de los usuarios y del servidor. Los datos de registro de IIS pueden ayudar a regular el acceso al contenido, evaluar la popularidad del contenido, planear los requisitos de seguridad y resolver problemas potenciales en los sitios Web o FTP. El registro de actividad del sitio IIS no se debe confundir con el registro de sucesos efectuado por Windows XP ó 2000, que se muestra con el Visor de sucesos. El registro en IIS es más extenso y lo veremos a continuación.

El proceso de registro

El registro de un sitio Web o FTP se realiza mediante unos módulos que funcionan independientemente de las demás actividades del servidor. Podemos elegir el formato de los registros para cada sitio Web o FTP individual. Si está habilitado el registro en un sitio, podemos habilitarlo o deshabilitarlo individualmente para cada uno de sus directorios.

Cada formato de registro utiliza una zona horaria diferente como base para las horas mostradas en los registros. El formato extendido W3C utiliza el Horario universal coordinado (UTC), lo que antes llamábamos hora del meridiano de Greenwich, Los otros formatos utilizan la hora local. Las horas mostradas en los archivos de registro reflejan la hora que el servidor utiliza para procesar las peticiones y las respuestas. Estas horas no reflejan el tiempo transcurrido en la red hasta llegar al cliente ni el tiempo de proceso del cliente.

Formatos de archivo de registro

Podemos elegir el formato que el servidor Web utiliza para registrar la actividad de los usuarios. Disponemos de los siguientes formatos:

1. Formato de archivo de registro extendido W3C
2. Formato de registro de Microsoft IIS
3. Formato del archivo de registro común NCSA

El formato de archivo de registro extendido W3C, el formato de archivo de registro Microsoft IIS y el formato de archivo de registro NCSA son todos formatos de texto ASCII. El formato extendido W3C y el formato NCSA registran datos con formato de año de cuatro dígitos. El formato de Microsoft IIS utiliza un formato de dos dígitos para el año 1999 y anteriores y un formato de cuatro dígitos para los años posteriores. El formato de registro que se proporciona con Microsoft IIS asegura la compatibilidad con versiones anteriores de IIS. Únicamente se puede utilizar el formato de archivo de registro extendido W3C para crear formatos de registro personalizados con los campos precisos que se necesiten.

A título informativo comentaremos los tres tipos de registro que existen en IIS:

1. Formato de archivo de registro extendido W3C

El formato extendido W3C es un formato ASCII que puede personalizarse con diversos campos diferentes. Puede incluir campos que considere importantes y limitar al mismo tiempo el tamaño del registro si omite los campos que no desea. Los campos están separados por espacios. La hora se registra como UTC (Horario universal coordinado).

En el ejemplo siguiente se muestran líneas de un archivo que incluye los campos siguientes: Hora, Dirección IP del cliente, Método, Recurso (URL) visitado, Estado del protocolo y Versión del protocolo.

```
#Software: Servicios de Internet Information Server 5.1 de Microsoft
#Versión: 1.0
#Fecha: 1998-05-02 17:42:15
#Campos: time c-ip cs-method cs-uri-stem sc-status cs-version
17:42:15 172.16.255.255 GET /default.htm 200 HTTP/1.0
```

La entrada anterior indica que el 2 de mayo de 1998, a las 5:42 p.m., UTC, un usuario con HTTP versión 1.0 y dirección IP 172.16.255.255 emitió un comando GET de HTTP para el archivo \Default.htm. La petición se resolvió sin errores. El campo #Fecha: indica cuándo se hizo la primera entrada de registro, que es cuando se creó el registro. #Versión: indica que se utilizó el formato de registro W3C.

Podemos seleccionar cualquiera de los campos, pero puede que algunos no tengan información disponible para algunas peticiones. Para aquellos campos seleccionados que no tengan información aparecerá un guión (—) en el campo como marcador de posición.

Al seleccionar este formato hemos dicho que podemos personalizar sus campos. Si pulsamos en el botón "Propiedades" nos aparecerá una pantalla cuya segunda ficha será como esta:

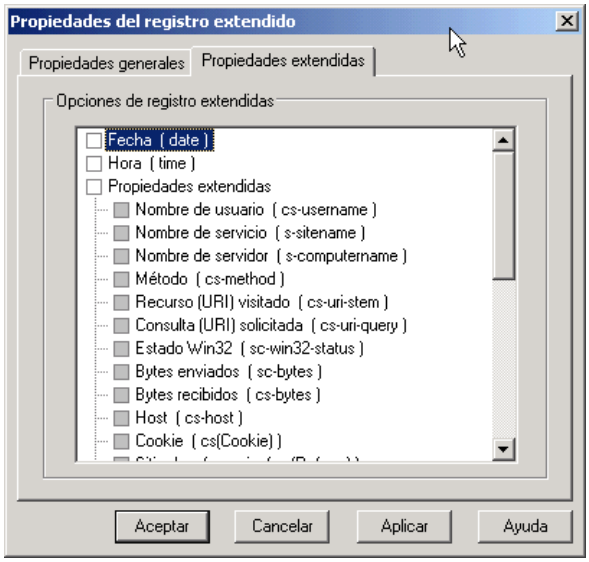

Figura 17 Propiedades del Registro Extendido

Esta pantalla permite indicar que campos queremos que figuren en el registro.

2. ***Formato de registro de Microsoft IIS***

El formato de Microsoft IIS es un formato ASCII fijo (no puede personalizarse) pero registra más datos que el formato común NCSA. El formato de Microsoft IIS incluye elementos básicos como la dirección IP del usuario, el nombre de usuario, la fecha y la hora de petición, el código de estado de servicio y el número de bytes recibidos. Además, incluye elementos detallados como el tiempo transcurrido, el número de bytes enviados, la acción (por ejemplo, una descarga realizada con un comando GET) y el archivo de destino. Los elementos se separan con comas, por lo que leer el formato resulta más sencillo que con los demás formatos ASCII, que utilizan espacios como separadores. La hora de registro es la local.

Al abrir un archivo con formato Microsoft IIS en un editor de textos, las entradas serán similares a las de los ejemplos siguientes:

192.168.114.201, —, 03/20/98, 7:55:20, W3SVC2, VENTAS1, 192.168.114.201, 4502, 163, 3223, 200, 0, GET, /DeptLogo.gif, —,
172.16.255.255, anónimo, 03/20/98, 23:58:11, MSFTPSVC, VENTAS1, 192.168.114.201, 60, 275, 0, 0, 0, PASS, /intro.htm, —,

En las tablas siguientes se interpretan las entradas anteriores. La fila superior de cada tabla proviene de la segunda instancia del sitio Web (que aparece en "Servicio" como W3SVC2) y la fila inferior de la primera instancia del sitio FTP (que se indica en

"Servicio" como MSFTPSVC1). El ejemplo se presenta en tres tablas por la limitación de ancho de página.

Dirección IP del usuario	Nombre del usuario	Fecha	Hora	Servicio instancia	Uso y resultado
192.168.114.201	—	03/20/98	7:55:20	W3SVC2	SALES1
172.16.255.255	anonymous	03/20/98	23:58:11	MSFTPSVC1	SALES1

Dirección IP del servidor	Tiempo empleado	Bytes enviados	Bytes recibidos	Código de estado de servicio	Código de estado de Windows
192.168.114.201	4502	163	3223	200	0
172.16.255.255	60	275	0	0	0

Tipo de solicitud	Destino de la operación	Parámetros
GET	/DeptLogo.gif	—
[376] PASS	/intro.htm	—

Figura 18 Entradas de los archivos logs

En el ejemplo, la primera entrada indica que un usuario anónimo, con la dirección IP 192.168.114.201, envió un comando GET de HTTP para el archivo de imagen /DeptLogo.gif a las 7:55 a.m. del 20 de marzo de 1998, desde un servidor llamado VENTAS1 que tiene la dirección IP 172.21.13.45. La petición HTTP de 163 bytes ha tenido un tiempo de proceso de 4502 milisegundos (4,5 segundos) y devolvió, sin errores, 3223 bytes de datos al usuario anónimo.

En el archivo de registro, todos los campos terminan en coma (,). Un guión (—) actúa como marcador de posición si no hay un valor válido para un campo determinado.

3. ***Formato del archivo de registro común NCSA***

El formato común NCSA es un formato ASCII fijo (no puede personalizarse), disponible para sitios Web, pero no para sitios FTP. Registra información básica acerca de las peticiones de los usuarios, como nombre de host remoto, nombre de usuario, fecha, hora, tipo de petición, código de estado HTTP y número de bytes enviados por el servidor. Los elementos están separados con espacios en blanco y la hora de registro es la local.

Al abrir un archivo con formato común NCSA en un editor de textos, las entradas serán similares a las del ejemplo siguiente:

172.21.13.45 — REDMOND\fred [08/Apr/1997:17:39:04 -0800] "GET /scripts/iisadmin/ism.dll?http/serv HTTP/1.0" 200 3401

Nota En la entrada anterior, el segundo campo (que mostraría el nombre de inicio de sesión remota del usuario) está vacío y está representado por el guión que sigue a la dirección IP 172.21.13.45.

En las tablas siguientes se interpreta la entrada de ejemplo anterior. Se utilizan dos tablas por la limitación de ancho de página.

Nombre de host remoto	Nombre de inicio de sesión remota	Nombre de usuario	Fecha	Hora y diferencia con GMT
172.21.13.45	—	Rioja\jose	08 de abril de 1998	17:39:10 -0800

Petición/Versión	Código de estado de servicio	Bytes enviados
GET /scripts/iisadmin/ism.dll?http/serv HTTP/1.0	200	3401

Figura 19 Limitación para ancho de página

La entrada indica que un usuario llamado José del dominio Rioja, con la dirección IP 172.21.13.45, envió un comando GET de HTTP (es decir, descargó un archivo) a las 5:39 p.m. del 8 de abril de 1998. La petición devolvió, sin errores, 3401 bytes de datos al usuario José.

Tamaño de archivo de registro y creación de nuevos archivos de registro

Cuando está habilitado el registro de IIS, (lo está de manera predeterminada) se generan nuevas entradas de registro siempre que un usuario tiene acceso al servidor. Esto produce un incremento progresivo del tamaño del archivo de registro o del número de archivos de registro. Podemos necesitar equilibrar la recopilación de datos detallados con la necesidad de limitar los archivos a un número y tamaño fáciles de administrar. IIS ofrece dos opciones para administrar la generación de datos de registro y la creación de nuevos archivos de registro.

Una forma de administrar los datos de registro es personalizar el registro extendido W3C de modo que sólo se recopilen los datos que se necesitan. Otra opción para administrar archivos de registro es limitar el tamaño del registro mediante el cambio de la frecuencia de creación del archivo de registro. Esta opción aparece pulsando el botón Propiedades:

Figura 20 Propiedades del registro extendido

Los archivos de registro son simplemente archivos ASCII (de texto). Si ha creado muchos archivos pequeños y prefiere uno grande, puede combinarlos como haría con cualquier archivo ASCII.

Si el servidor se queda sin espacio en disco cuando IIS intenta agregar una entrada de registro a un archivo, el registro de IIS se cierra. Al mismo tiempo, en el registro de aplicación del Visor de sucesos de Windows, se registra un suceso. Cuando vuelve a haber espacio disponible en disco, se reanuda el registro de IIS. Esto hace que se registre un suceso adicional en el registro de aplicación del Visor de sucesos de Windows.

Nombres de archivo de registro

En los nombres de los archivos de registro se utilizan las primeras letras para representar el formato y los números restantes para indicar el marco horario o la secuencia del registro. Consulta la tabla siguiente para obtener más información. Las letras en cursiva representan dígitos: nn para dígitos secuenciales, aa para el año, mm para el mes, ss para la semana del mes, dd para el día y hh para la hora en formato de 24 horas (es decir, 17 es 5:00 p.m.).

Formato	Criterio para nuevo registro	Patrón de nombre de archivo
Formato de registro de Microsoft IIS	Por el tamaño del archivo	inetsvnn.log
	Cada hora	inaammddhh.log
	Diario	inaammdd.log
	Semanal	inaammss.log
	Mensual	inaamm.log
Formato del archivo de registro común NCSA	Por el tamaño del archivo	ncsann.log
	Cada hora	ncaammddhh.log
	Diario	ncaammdd.log
	Semanal	ncaammss.log
	Mensual	ncaamm.log
Formato de archivo de registro extendido W3C	Por el tamaño del archivo	extendnn.log
	Cada hora	exaammddhh.log
	Diario	exaammdd.log
	Semanal	exaammss.log
	Mensual	exaamm.log

Figura 21 Formatos de los Archivos de IIS

3.2.3 Tunning

Administradores de servidores Web supervisan a menudo sus servidores Web para crear una línea de fondo de su actuación. Una línea de fondo de la actuación es una colección de datos que indican cómo los servidores están realizando cuando todo está corriendo fácilmente. Antes de que hiciera los cambios a sus servidores en un ambiente de la producción (por ejemplo. rodando fuera una nueva versión de una aplicación de Web o hardware cambia) administradores ponen a punto sus servidores en un ambiente de prueba lograr la línea de fondo de la actuación establecida. Poniendo a punto sus servidores de esta manera, ellos aumentan al máximo el canal de salida y minimizan la contestación de una aplicación Web que crea una experiencia buena para clientes que acceden a sus servidores Web.

La Condensación habilitada de HTTP (IIS 6.0)

Usted puede habilitar la condensación de HTTP un ancho de servidor o en un directorio específico. La condensación de HTTP mejora utilización del ancho de banda y velocidades a la actuación de un sitio Web. Además, usted puede comprimir las contestaciones estáticas o dinámicas.

Para usar el ancho de banda disponible más eficazmente, permítale la condensación de HTTP a IIS. La condensación de HTTP proporciona el tiempo de la transmisión más

rápido entre los navegadores habilitando condensaciones en IIS, sin tener en cuenta si su volumen se sirve de almacenamiento local o un recurso de UNC. Usted puede comprimir archivos estáticos y archivos de contestación de aplicación. Comprimiendo los archivos de contestación de aplicación normalmente se llama la condensación dinámica.

Importante

Usted debe ser un miembro de los Administradores agrúpese en la computadora local para realizar el procedimiento siguiente o procedimientos. Como una mejor práctica de seguridad, ingrese a su computadora usando una cuenta que no está en los administradores, agrúpese, y entonces use las rutas para ejecutar a administrador de IIS como un administrador. A una sugerencia del orden, rutas del tipo: Administrative_AccountName"mmc% systemroot%\system32\inetsrv\iis.msc."

Habilitando mantener el tiempo de vida en HTTP (IIS 6.0)

Más navegadores web demandan que la subsistencia de la conexión del servidor con el cliente se abre mientras el servidor envía los elementos múltiples (Archivos .htm, .gif y jpg) al cliente, esto se podría mejorar guardando la conexión del cliente llamado un HTTP Keep-Alive. Guardar vivo es una especificación de HTTP que mejora la actuación del servidor. Mantener el tiempo de vida HTTP se habilita por defecto.

Importante

Usted debe ser un miembro de los Administradores agrúpese en la computadora local para realizar el procedimiento siguiente o procedimientos. Como una mejor práctica de seguridad, ingrese a su computadora usando una cuenta que no está en los administradores, agrúpese, y entonces use las rutas para ejecutar a administrador de IIS como un administrador. A una sugerencia del orden, rutas del tipo: Administrative_AccountName"mmc% systemroot%\system32\inetsrv\iis.msc."

HTTP habilitando mantener el tiempo de vida para guardar las Conexiones abiertas (IIS 6.0)

Un navegador hace las demandas múltiples típicamente para transmitir una página Web entera. Para reforzar la actuación del servidor, más navegadores web demanda que la subsistencia de la conexión abierta del servidor por estas demandas múltiples son un rasgo conocido como HTTP Keep-Alives.

Sin HTTP Keep-Alives, un navegador que hace las numerosas demandas para una página que contiene los elementos múltiples, como los gráficos, podría requerir una conexión separada para cada elemento. Estas demandas adicionales y conexiones requieren la actividad del servidor extra y recursos, la eficacia del servidor decreciente. Las conexiones adicionales también hacen un navegador más lento y menos sensible, sobre todo por una conexión lenta.

HTTP Keep-Alives se habilita por defecto en IIS 6.0 que obedece la especificación de HTTP/1.1 para HTTP Keep-Alives. IIS sostiene una conexión inactiva con tal de que la propiedad ConnectionTimeout de la metabase especifique (el valor predefinido es 120 segundos).

Si usted desactiva HTTP Keep-Alives, el servidor ignora un clientes piden guardar la conexión abierta. Por consiguiente, desactive HTTP Keep-Alives sólo por una razón específica y si usted entiende claramente cómo este cambio afecta su servidor.

HTTP Keep-Alives se requiere para la seguridad integrada o la autenticación conexión basada en la autenticación Integrada de Windows. Si usted desactiva HTTP Keep-Alives para los sitios de Web que usan la autenticación integrada de Windows.

Las Interrupciones de Conexión (IIS 6.0)

Los tiempos exteriores de conexión ayudan a que se reduzca la pérdida de procesar los recursos consumidos por las conexiones ociosas. Cuando usted habilita los tiempos exteriores de conexión, IIS da fuerza a los tiempos exteriores al nivel de conexión.

Importante

Usted debe ser un miembro de los Administradores agrúpese en la computadora local para realizar el procedimiento siguiente o procedimientos. Como una mejor práctica de seguridad, ingrese a su computadora usando una cuenta que no está en los administradores, agrúpese, y entonces use las rutas para ejecutar a administrador de IIS como un administrador. A una sugerencia del orden, rutas del tipo: Administrative_AccountName"mmc% systemroot%\system32\inetsrv\iis.msc."

Limitando las Conexiones (IIS 6.0)

Los límites de conexión restringen el número de conexiones del cliente simultáneas a sus sitios Web y su servidor Web. La memoria conserva las conexiones limitando y

protegiendo contra los ataques malévolos, aunque estas estén diseñadas para cargar excesivamente su servidor Web con los miles de demandas del cliente.

Otra manera de limitar las conexiones a su servidor Web ha terminado ahogando el ancho de banda.

Importante

Usted debe ser un miembro de los Administradores agrúpese en la computadora local para realizar el procedimiento siguiente o procedimientos. Como una mejor práctica de seguridad, ingrese a su computadora usando una cuenta que no está en los administradores, agrúpese, y entonces use las rutas para ejecutar a administrador de IIS como un administrador. A una sugerencia del orden, rutas del tipo: Administrative_AccountName"mmc% systemroot%\system32\inetsrv\iis.msc."

Limitando las Conexiones para Manejar los Recursos (IIS 6.0)

Los límites de conexión restringen el número de conexiones simultáneas a sus sitios Web y su servidor Web. Si el número de alcances de conexiones máximo que repartió, toda la conexión subsecuente intenta el retorno de un error y entonces estará desconectado.

Limitando las conexiones, usted puede conservar el ancho de banda para otros usos, como los servidores del correo electrónico, servidores de las noticias, u otro sitio Web que corren en la misma instalación. Limitando las conexiones también la memoria de las conservas se protege contra los ataques malévolos.

Para determinar si usted necesita limitar las conexiones, use al Amonestador del Sistema para anotar las Conexiones Actuales, Conexiones Máximas, y los contadores de Esfuerzos de Conexión Totales en el Web Service y FTP Service los objetos. Continúe anotando hasta que usted tenga un buen sentido del rango normal; típicamente, anotando pueden tomar varios días a una semana o más y debe repetirse a los intervalos regulares.

Usted puede establecer reparar WWW globales o reparar los límites de conexión para todo el Web y sitios de FTP, o usted puede establecer los límites de conexión en Web individual o sitios de FTP. IIS verifica para un límite de conexión global antes de verificar para la conexión limita en los sitios individuales. Si el límite de conexión

global se excede, IIS devuelve mensaje 403.9 al error (Prohibido - los Demasiados Usuarios) sin tener en cuenta el límite de conexión puesto para los sitios individuales. (Usted puede personalizar la conexión límite; sin embargo, IIS 6.0 no ofrece esa opción.)

El Uso de Memoria controlando (IIS 6.0)

Para ejecutar IIS 6.0 con el Servidor de Windows 2003, un servidor de Web especializado debe tener un mínimo de 128 MB de RAM. Sin embargo, su servidor podría exigir a más RAM ocuparse de las necesidades del recurso de aplicaciones personalizadas. Si sus aplicaciones personalizadas requieren cantidades grandes de memoria, proporcione 256 MB o más de RAM. La memoria adicional es particularmente beneficiosa para los sitios de comercio, los sitios con mucho volumen, y sitios con un volumen alto de tráfico.

3.2.4 Requisitos del Sistema

Requerimientos Hardware

Requerimientos Mínimos

Ordenador/Procesador	Se requieren 133 MHz, como mínimo; procesador de las familias Intel Pentium/Celeron, AMD K6/Athlon/Duron u otro compatible (Windows Server 2003 Standard Edition admite hasta cuatro CPU en un servidor)
Memoria	Se requieren 128 MB como mínimo (4 GB de RAM, como máximo)
Disco Duro	De 1,25 a 2 GB de espacio disponible en el disco duro
Disquetera	Unidad de CD-ROM o DVD-ROM
Monitor	Se requiere VGA o hardware que permita la redirección de la consola
Periféricos	Teclado y Microsoft Mouse o compatible, o hardware que permita la redirección de la consola
Otros	Para el acceso a Internet: –Algunas funciones de Internet pueden requerir acceso a Internet, una cuenta de Microsoft Passport y el pago de una cuota adicional a un proveedor de servicios; se pueden aplicar tarifas de llamadas de larga distancia

	–Módem de alta velocidad o conexión de banda ancha a Internet Para la red: –Adaptador de red apropiado para el tipo de red de área local, de área extensa, inalámbrica o doméstica a la que desee conectarse, y acceso a una infraestructura de red apropiada; el acceso a redes de terceros puede requerir tarifas adicionales

Requisitos recomendados

Ordenador/Procesador	Se recomienda un equipo con 550 MHz o una velocidad de reloj superior; procesador de las familias Intel Pentium/Celeron, AMD K6/Athlon/Duron u otro compatible (Windows Server 2003 Standard Edition admite hasta cuatro CPU en un servidor)
Memoria	Se recomiendan 256 MB de memoria RAM o más (máximo: 4 GB)
Disco Duro	De 1,25 a 2 GB de espacio disponible en el disco duro
Disquetera	Unidad de CD-ROM o DVD-ROM
Monitor	Se recomienda un monitor Super VGA (800 x 600) o con una resolución mayor
Periféricos	Teclado y Microsoft Mouse o compatible, o hardware que permita la redirección de la consola
Otros	Para el acceso a Internet: –Ciertas funciones de Internet pueden requerir acceso a Internet, una cuenta de Microsoft Passport y el pago de una cuota adicional a un proveedor de servicios; se pueden aplicar tarifas de llamadas de larga distancia –Módem de alta velocidad o conexión de banda ancha a Internet Para la red: –Adaptador de red apropiado para el tipo de red de área

	local, de área extensa, inalámbrica o doméstica a la que desee conectarse, y acceso a una infraestructura de red apropiada; el acceso a redes de terceros puede requerir tarifas adicionales

Tabla 12 Tabla de Requermientos Hardware

Requerimientos Software

Como requisito Software II 6.0 viene incluido en el Sistema Operativo Microsoft® Windows® Server 2003 Standard Edition, Windows® Server 2003, Enterprise y Windows® Servidor 2003 Datacenter.

Ranking de servidores por número de visitas y tabla estimada de visitas a servidores Web.

Para adentrarnos en estos términos primero se verán algunos conceptos que serán de mucha importancia para esclarecer cualquier duda.

3.3 Parámetros para la Optimización del Servidor Web.

Al hablar de optimización se hable de calidad, y la calidad se determina de acuerdo desempeño que en este caso hace referencia al desempeño del Servidor web.

Los parámetros que se han considerado para la optimización son los siguientes:

- funcionamiento
- Rendimiento
- Seguridad
- Aceptabilidad

Funcionamiento

Cuando se habla del funcionamiento del Servidor Web, nos referimos a todo el proceso que se realiza desde la instalación hasta la puesta en marcha del mismo.

En otras palabras, el funcionamiento se refiere a la puesta a punto del servidor. Para poner a punto a un servidor se debe tener encuenta lo siguiente: necesidad, cantidad de información a manejar, cantidad de trafico (peticiones y respuestas), servicios.

Rendimiento.

Significa que soporta más para lo cual es utilizado, mejorar el rendimiento del Servidor Web significa, configurarlo para mejorar su trabajo: Entrega a tiempo de resultados (Tiempos de respuestas), facilite la entrega de información de una manera más adecuada (entendible al administrador), mejora de los servicios.

Seguridad.

Sabemos que a través del Internet viaja mucha información, información que puede ser capturada por cualquier persona. Imaginémonos que esta información es confidencial y para enviarla tenemos que estar seguros de que nadie la conozca. Si bien es cierto, hoy en día los lenguajes de programación dan muchos servicios para hacer que la información que viaja a través de la red sea segura, también es cierto que el Servidor Web que estamos utilizando nos de el soporte necesario para asegurar la seguridad de nuestra información.

Aceptabilidad.

Cuando aceptamos algo es cuando ese algo satisface nuestra necesidad. Precisamente que un servidor sea aceptable es cuando nos garantiza que su funcionamiento, rendimiento y seguridad son los adecuados para satisfacer nuestras necesidades.

CAPITULO IV

4.1 DESARROLLO DE UNA GUIA REFERENCIAL PARA OPTIMIZACION DE SERVIDORES WEB

4.1.1 Apache

4.1.1.1 FUNCIONAMIENTO.

El funcionamiento del servidor Apache se refiere dejar listo el servidor para su funcionamiento. Cuando se dice que un servidor está listo es cuando está instalado y configurado de acuerdo a las necesidades.

En esta primera parte vamos abarcar desde la instalación hasta las configuraciones básicas para que el Servidor Apache funcione.

Instalación de Apache

Requerimiento previo a la instalación:

Los requerimientos previos a la instalación de apache son muy escuetos. Necesitará un mínimo de 12MB de espacio temporal en la unidad de disco duro para el proceso de instalación. Tras la instalación, Apache ocupa cerca de 3MB, además del espacio que se utilice para colocar el contenido web.

También necesitará un compilador ANSI-C. El compilador GNU C, que se conoce como GCC, es el compilador recomendado, pero otros compiladores también trabajan bien si son compatibles con ANSI-C.

Perl es necesario en algunos scripts de soporte opcional. El soporte para Objetos Compartidos Dinámicos (DSO) está recomendada pero no es necesario.

Instalación

1. Se descarga el Servidor Apache desde la Página http://www.apache.org, esta es

una página propia de apache, en donde podemos encontrar desde los archivos fuentes del servidor, hasta trucos y la manera de cómo poder mejorar nuestro servidor web.

2. Una ves descargado se procede a instalar y las instrucciones son las siguientes:

#tar -zxvf /usr/local/apache/apache_2.0.52.tar.gz
#cd /usr/local/apache/apache_2.0.52
./configure –prefix=/usr/local/apache
make
make install
/usr/local/apache/bin/apachectl start

Puede cambiar el prefijo si desea instalarlo en otro lugar que no sea /usr/local/apache.

***Configurar Apache con APACI*.**

Apaci es un programa para la configuración de Apache. Para ejecutar APACI, ejecute el script configure en la línea de comando con su lista de argumento. Las instrucciones anteriores ejecuta el mismo script.

A continuación hablaremos de algunos comandos que se deben ejecutar con APACI.

--show-layout
Esta opción es muy útil a la hora de mostrar donde acaban todos los archivos cuando se ejecuta make al final del proceso.
--prefix
La ubicación predeterminada de Apache que hay que instalar es /usr/local/apache. La opción –prefix le permite cambiar a otra ubicación. Las primera versión de Apache mantenían los archivos en /usr/local/etc/httpd, y posiblemente quiera colocar aquí los archivos por cuestiones de compatibilidad.
./configure –prefix=/usr/local/etc/httpd

--enable-module
Esta opción y su opción asociada –disable-module le permite activar y desactivar determinados modulos. Si su sistema operativo soporta DSO (Objetos Compartidos

Dinámicos), probablemente use –enable-module con la opción –enable-shares para compilar ese modulo con un DSO.

Para instalar el modulo mod_speling, por ejemplo, tendríamos que utilizar la siguiente línea de comando:

. /configure –enable-module=speling –enable-shared=speling

4.1.1.2 Configuración de Rendimiento y Seguridad

Muchas veces cuando se instala Apache solo se deja en las configuraciones que vienen por defecto y solo se cambia: host virtuales, puerto, servername, documentroot y serveradmin. Claro que con esta mínima configuración el servidor Apache funciona bien, ya que los parámetros de rendimiento-seguridad no sólo tiene que ver con el servidor sino también con el sistema operativo.

Apache es un servidor general de web, que está diseñado para ser primero correcto y después rápido. Aún así, su rendimiento es bastante satisfactorio. Muchos sitios tienen menos de 10 Mbits de ancho de banda de salida, que Apache puede ocupar usando tan sólo un servidor basado en un Pentium de gama baja. En la práctica, los sitios con más ancho de banda requieren más de una máquina para ocuparlo debido a otras restricciones (como sobrecarga por CGI o transacciones de bases de datos). Por esos motivos, el enfoque del desarrollo ha sido en mayor medida para la correctitud y configurabilidad.

Desafortunadamente, mucha gente pasa por alto esos hechos y cita números brutos de rendimiento como si fueran indicativos de la calidad de un servidor web. Hay un mínimo rendimiento básico que es aceptable, por encima del cual la velocidad adicional sólo abastece a un segmento mucho más reducido del mercado.
A continuación mostraremos una configuración para mejorar al servidor Apache en los parámetros de rendimiento-seguridad.

Aunque el rendimiento y la seguridad son parámetros muy importante al momento de configurar cualquier servidor, también es cierto que son contradictorios, ya que no se puede garantizar al cien por ciento rendimiento y seguridad a la vez:

1. Realmente apache es igual a decir httpd.conf, ya que en este archivo se realiza

todas las configuraciones para que el servidor funcione de acuerdo a nuestras necesidades. Una recomendación, siempre que se realice algún cambio significativo al archivo httpd.conf se debe antes realizar una copia de este archivo por si algo sale mal.

Para conocer como está configurado nuestro servidor web, debemos abrirlo, y lo hacemos con el editor de texto de nuestra preferencia. Recomendamos que para realizar cualquier configuración la realicemos mediante el editor *vi*, ya que este editor nos permite distinguir a través de colores: directivas, comentarios, archivos, etc.

Para abrir el archivo httpd.conf lo hacemos de la siguiente manera:

#vi /etc/httpd/conf/httpd.conf

Esto dependerá mucho del lugar en donde hayamos instalado el Apache, la primera es cuando instalamos el Apache que viene con los instaladores del Linux o también cunado instalamos con el comando *yum*.Los otros ejemplos es cuando hacemos una instalación personalizada de Apache.En cualquiera de los casos aparecerá la siguiente ventana con las configuraciones por defecto del servidor Apache.

2. La primera directiva que vamos a ver es la directiva *ServerRoot* (*ver figura 23*) la cual establece el directorio en el que se van a almacenar los archivos del servidor. Se presupone que la mayoría de las directivas que especifican una ruta de un archivo o un directorio se den en relación con esta directiva, a menos que lleven una barra al principio. Los directorios que normalmente están ubicados en esta directiva son: conf/ y logs/.

ServerRoot "/etc/httpd"

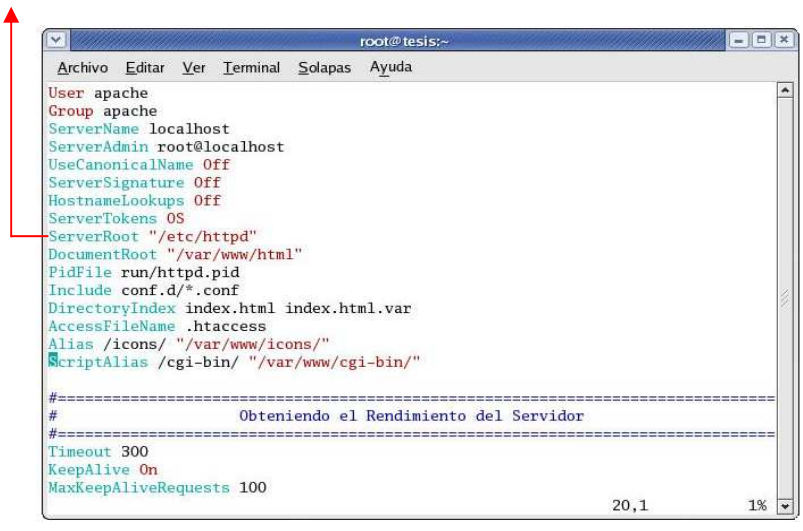

Figura 22: Especificación del Server

3. La siguiente directiva es *PidFile*, esta directiva especifica la ubicación del número PID (ID de proceso) del servidor. Su configuración se muestra en la figura 3.

   PidFile run/httpd.pid

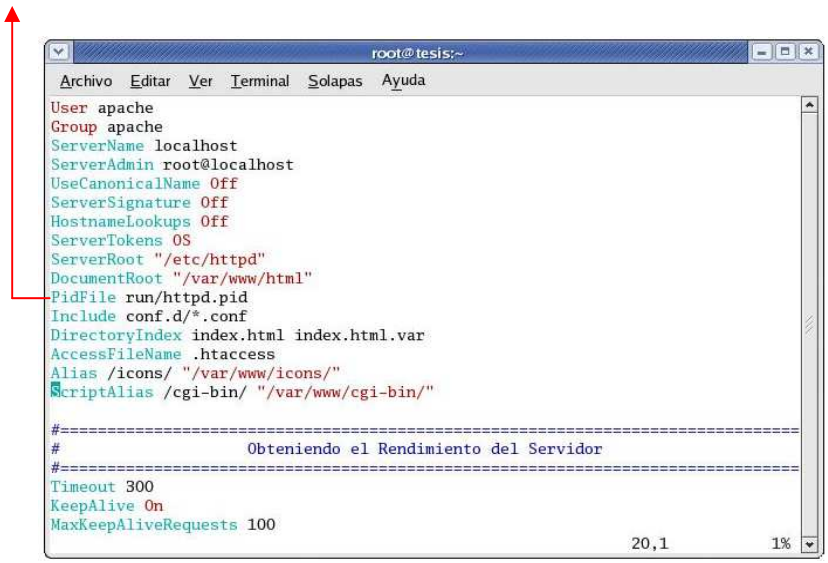

Figura2 3: Especificación del *PidFile*

4. La siguiente directiva es *Listen* y esta le indica al servidor que escuche las solicitudes de más de una dirección IP y/o puerto TCP/IP. Por defecto, apache escucha en todas las interfaces de red, pero sólo en el puerto especificado por la directiva Port.

   Si se especifica un número de puertos sin la dirección IP, Apache escuchará en ese puerto todas las interfaces:

   listen 80

   listen 8081

   listen 1352

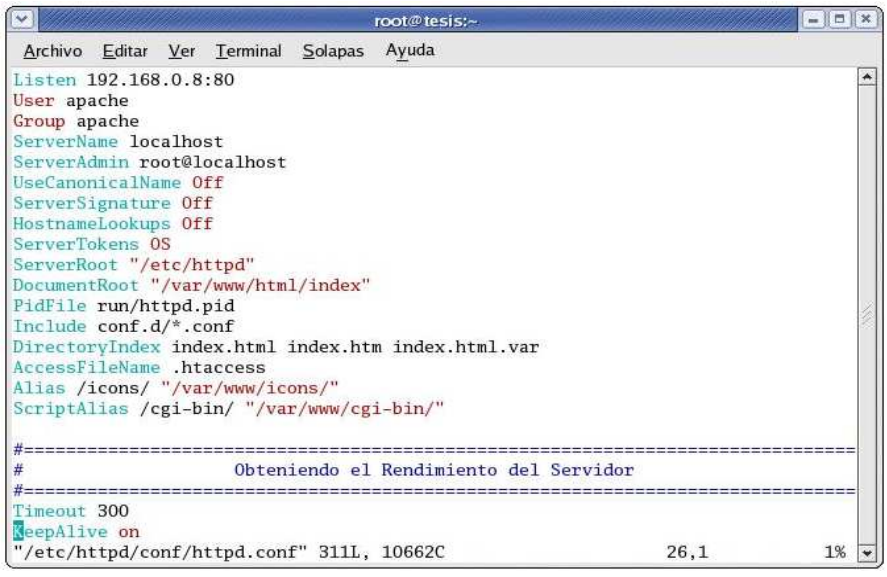

Figura 24: Muestra el archivo httpd.conf en el editor

Si se especifica además de una dirección IP, un puerto, Apache escuchará en esa combinación dirección-puerto especificado, así como se muestra en la figura 25.

Listen 127.0.0.1:80

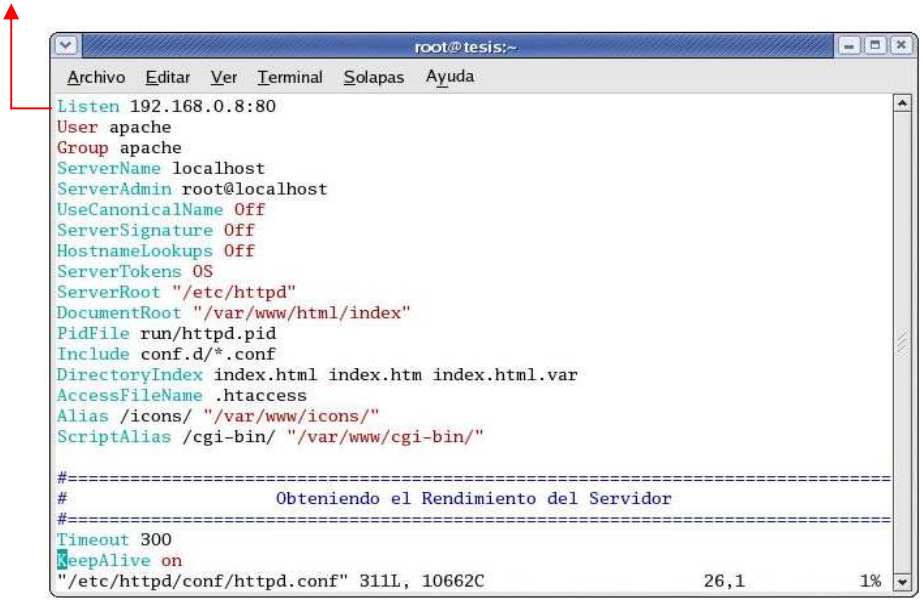

Figura 25 :Especificación *Listen*

5. Lo siguiente que se debe especificar es el directorio desde el cual se van a guardar los archivos HTML. Otros archivos que contengan archivos HTML puede ser configurado con la directiva *Alias*.

DocumentRoot "/var/www/html"

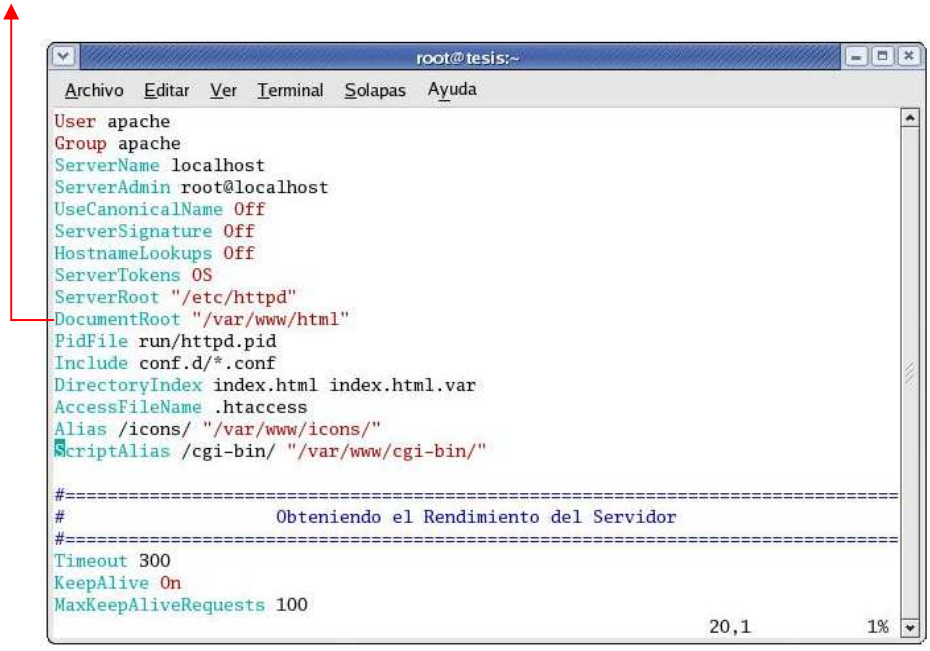

Figura 26: muestra la especificación del *DocumentRoot*

6. Luego se debe especificar la dirección de correo electrónico de la persona responsable del servidor. Esta dirección se incluye automáticamente en los

mensajes de error del servidor.

ServerAdmin jbajaÑa@espoch.edu.ec

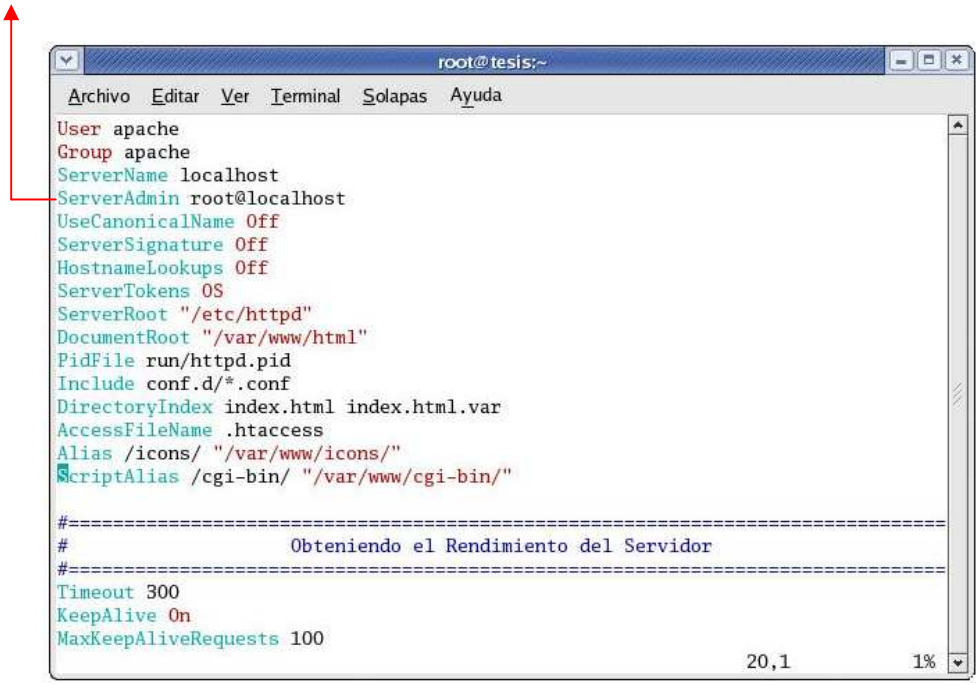

Figura 27: muestra la especificación del *ServerAdmin*

7. Luego se especifica el nombre de nuestro servidor (véase figura 7), que se usa cuando el servidor está construyendo direcciones de redireccionamiento. Si no se especifica esto, el servidor tratará de saberlo

   ServerName www.espoch.edu.ec

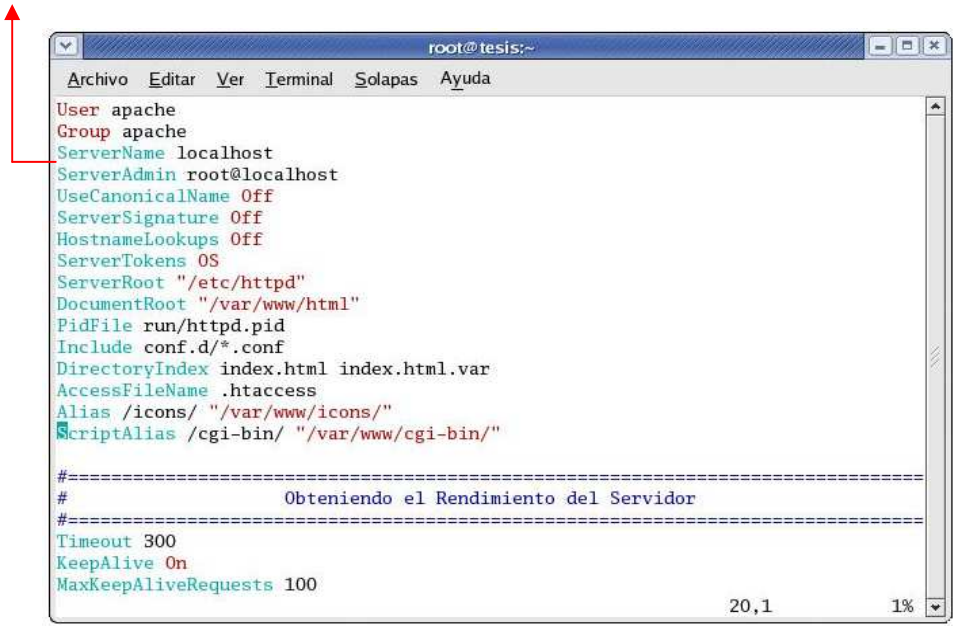

Figura 28: Especificación del *ServerName*

por sí msmo mediante búsqueda DNS. Sin embargo, es posible que no se consiga el

nombre deseado. La directiva quedaría de la siguiente manera. *ServerName* *www.espoch.edu.ec*

A continuación se muestra en la figura30 una explicación de lo que hace el servidor Web cuando no encuentra la especificación *ServerName*.

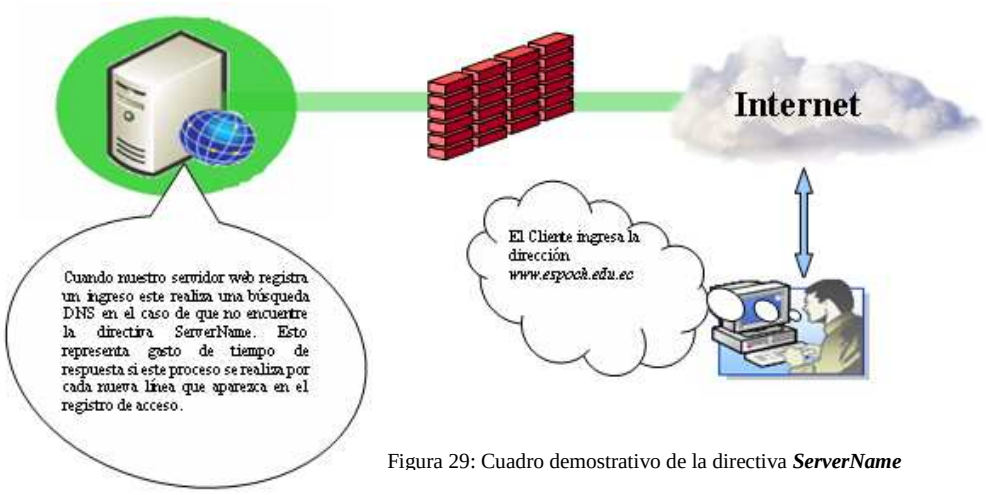

Figura 29: Cuadro demostrativo de la directiva ***ServerName***

8. Cuando el cliente hace una petición al servidor web comienza una conversación entre ambos. Esta conversación se realiza para establecer la comunicación entre los dos. La cual consiste en paquetes que son enviados por el cliente que le dicen al servidor que se desea establecer una conexión y el servidor le envía al cliente paquetes de aceptación o de negación para tal conexión. Esta conversación entre el cliente y el servidor debe tener un tiempo de duración. Ese tiempo de duración lo especificamos con la directiva *TimeOut*. Específicamente, en el momento que se produzca uno de estos eventos.

   - El tiempo total para recibir una solicitud GET.
   - El tiempo que hay entre los paquetes de una solicitud PUT o POST.
   - El tiempo que hay entre los ACK de transmisión de paquetes de respuesta.

   Veamos la instrucción:

   Timeout 120

   Esta instrucción suele estar especificada por defecto. Pero puede ser modificada por el administrador de acuerdo a su criterio.
9. Las siguientes instrucciones nos permiten acelerar los tiempos de respuesta. Esta mejora en el tiempo de respuesta lo hacemos con dos directivas. La primera es *keepAlive* y la segunda es *MaxKeepAliveRequests*, estas directiva deben trabajar

juntas para su mejor funcionamiento. La primera sirva para hacer varias solicitudes en una misma conexión, y la segunda define cuantas son las solicitudes que se van a servir por dicha conexión.

Entonces empezaremos por la primera:

→ Cuando un cliente hace una solicitud, este introduce un URL de la página que desea visualizar en el browser. Suponiendo que esta página tenga 5 (cinco) objetos, el cliente deberá realizar 6 (seis) solicitudes al servidor, una para la página y cinco por los cinco objetos, ósea, una para cada objeto. Esto indica que por cada solicitud debe establecer una conexión y realizar varios procesos para establecer dichas conexiones (ver figura 9), por lo que esto representaría demasiado tiempo de la CPU. Por ejemplo, si se tratase de 100 solicitudes a la misma página, esto significaría varios milisegundos ocupados en realizar las 600 conexiones. En otras palabras, sólo el proceso de petición de una página afectaría el rendimiento del servidor web en términos de: Tiempos de respuesta por la ocupación de la CPU en otros procesos.

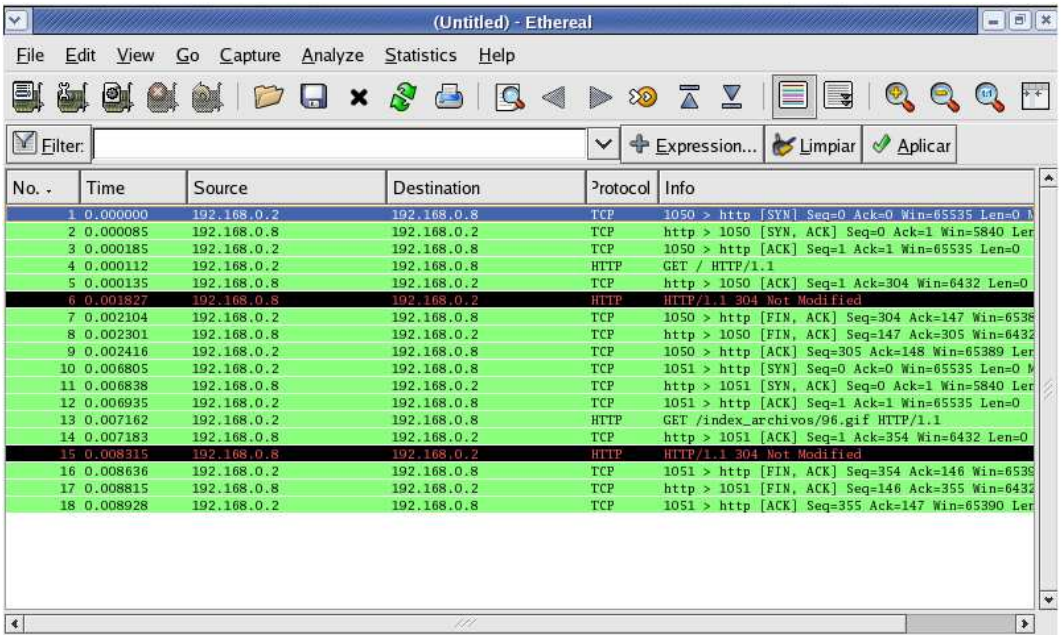

Figura 30: Muestra cada uno de los procesos en la petición de una página

Todo esto es muy normal que suceda en un servidor web, y no habría ningún problema cuando este servidor web tiene poca concurrencia. Pero si se tratase de un servidor que recibe muchas visitas, entonces esto significaría un problema al rendimiento del mismo.

Existe una instrucción que nos permite minimizar en gran medida esta situación. Esta instrucción es la siguiente: (ver figura 32).

KeepAlive on

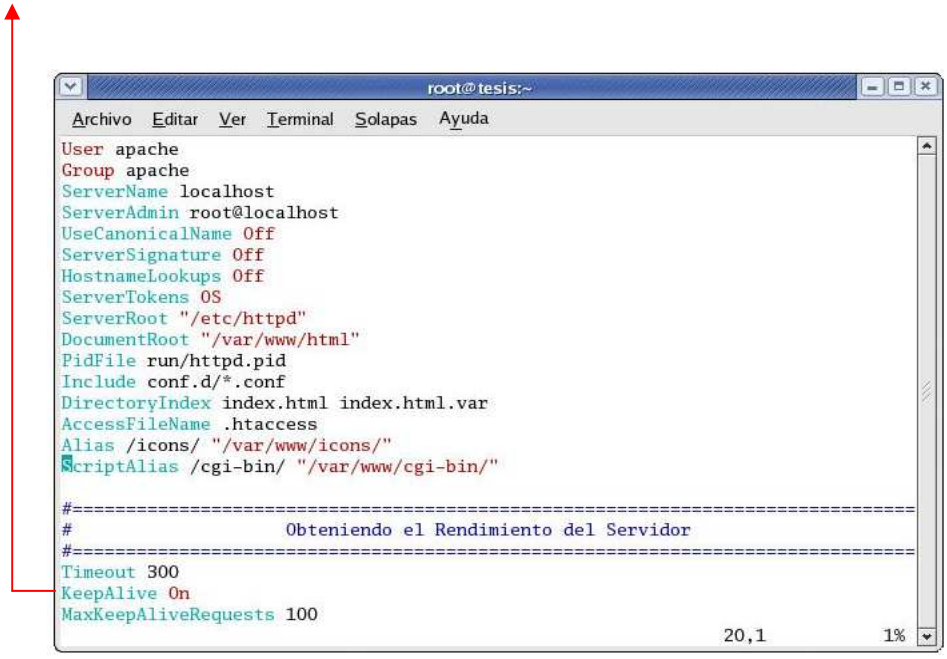

Figura 31: muestra la especificación del

Por lo regular esta directiva viene por defecto en *off* (desactivado) cuando se instala Apache. Nuestro trabajo es poner esta directiva en *on* (encendido como muestra la figura 10) para con esto lograr: tomando el ejemplo anterior, el cliente ya no deberá realizar una conexión por cada solicitud sino que establecerá una conexión por la cual se atenderán a todas las peticiones que se le hagan al servidor.

Esto quiere decir que encada línea (proceso) que aparece en la figura 9, irán al menos un proceso secundario. Esto depende mucho de cuantos procesos secundarios se requieran para realizar el proceso primario.

En la figura 33 podemos ver que cada proceso TCP posee varios procesos a la vez, comprobando con esto lo dicho anteriormente.

Cuando un proceso primario posee varios procesos secundarios, se produce una mejora ya que el de comunicación no está tan saturado por la gran cantidad de paquetes que transitan por este.

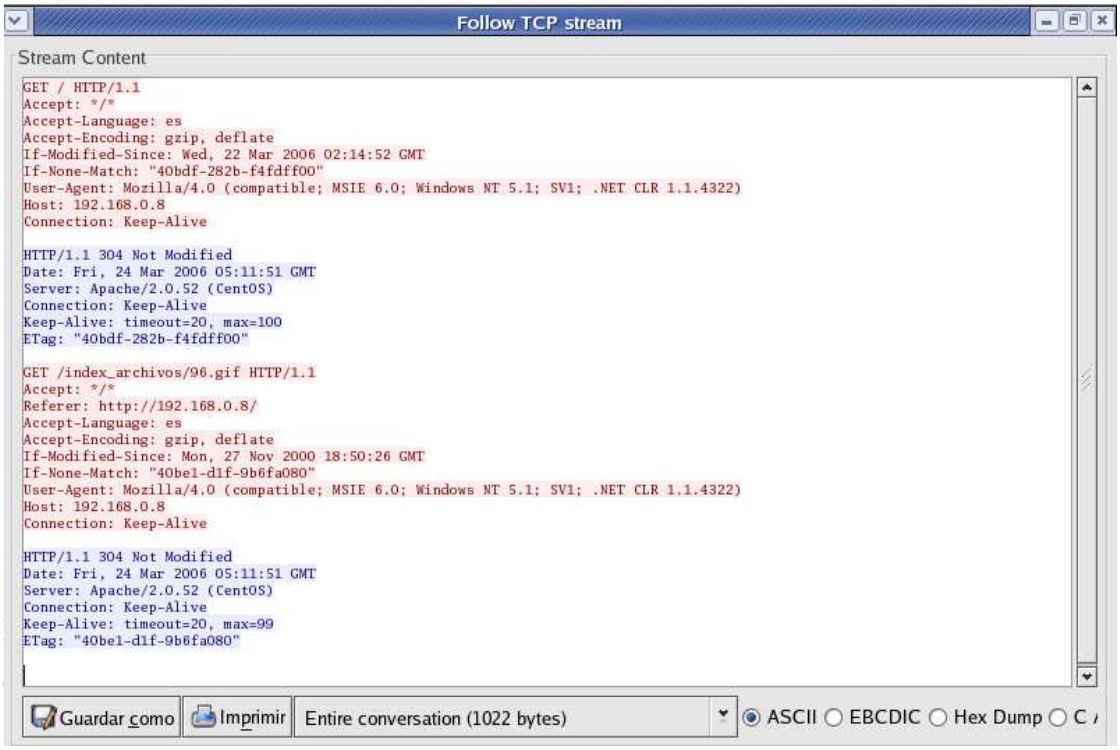

Figura 32: Podemos ver el contenido de uno de los procesos TCP

o

curre lo mismo cuando se deja en off a la directiva *KeepAlive*. La figura 31 muestra la cantidad de procesos que se realizan cuando esta directiva está en off, comparemos los resultados cuando de cambia el valor de esta directiva a on expuestos en la figura 34.

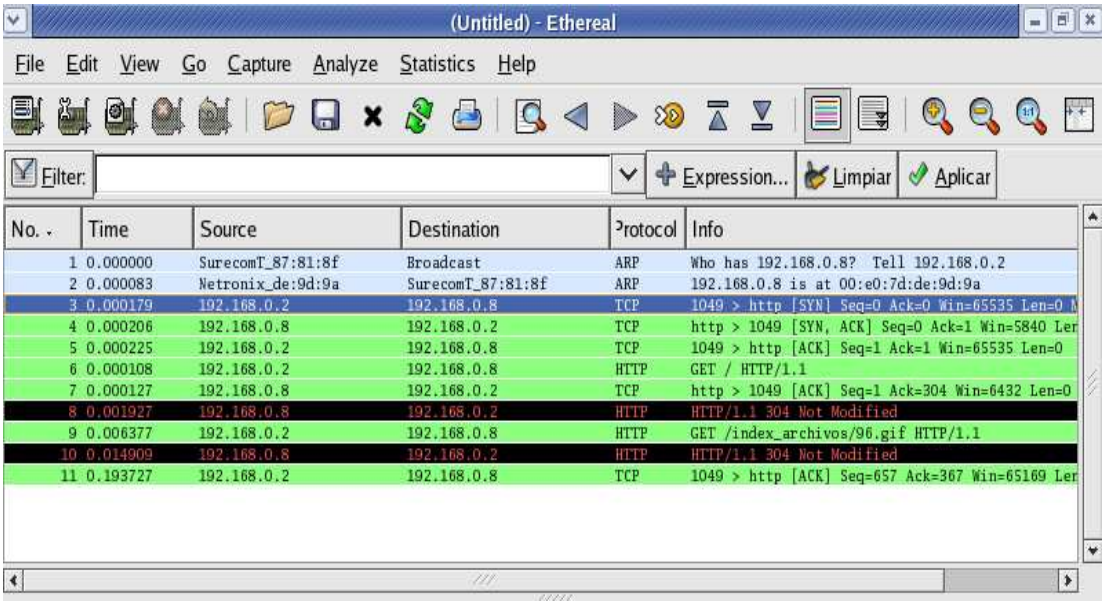

Figura 33: Muestra los resultados de capturar los paquetes
Cuando la directiva *KeepAlive* está en *on*

En la figura 35 veremos que sucede cuando se deja en off a la directiva *KeepAlive*.

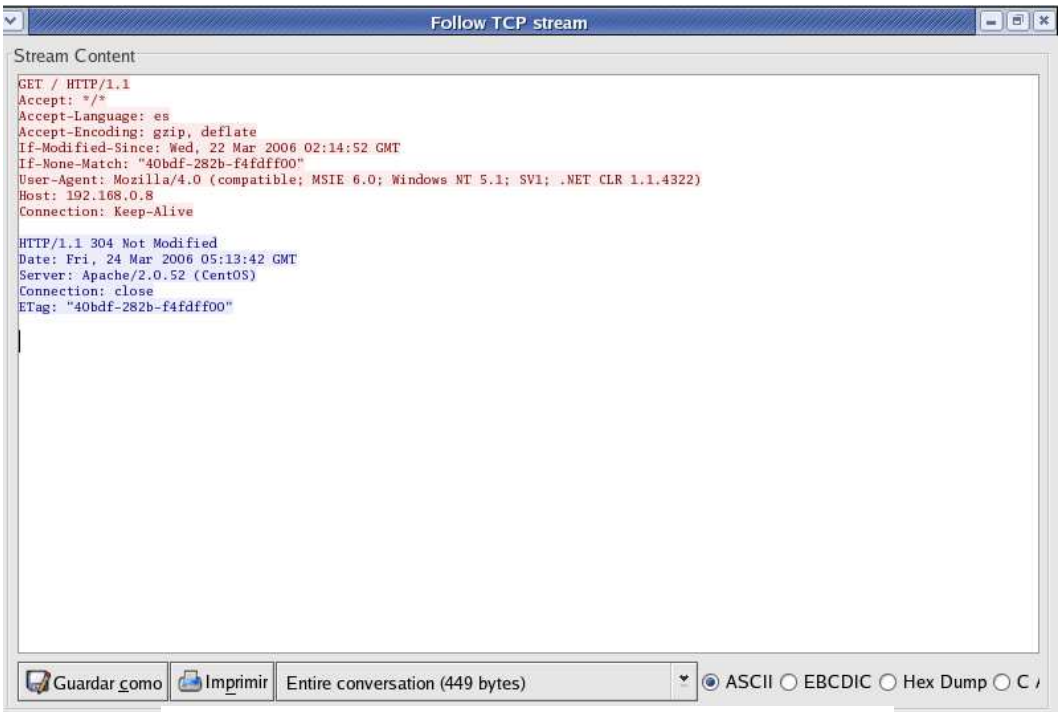

Figura 34: Proceso TCP cuando *KeepAlive* se encuentra en *off*

Esto nos ayudará a minimizar el número de conexiones y el número de procesos que se realizan cuando se va a establecer una conexión. Lo que significa que se liberará un poco a la CPU.

→ Determinar una conexión para atender a varias solicitudes resulta beneficioso para mejorar el rendimiento del servidor web. Con esto queda abierto un canal para que el cliente haga las peticiones que fueran necesarias al servidor, el servidor por su parte no sabrá cuantas son las solicitudes que deberá atender por esta conexión, provocando así que el servidor espere que se venza el tiempo de la conexión para cerrarla, sabiendo que una conexión se cierra cuando el cliente haya terminado de solicitarle todos los objetos al servidor o cuando haya expirado el tiempo de la conexión.

Para que el servidor funcione adecuadamente tenemos que especificarle cuantas son las peticiones que realizará por cada conexión y esto lo haremos con la siguiente directiva:

MaxKeepAliveRequests 100

Esta opción cuando se instala Apache viene con un valor 100 por defecto, pero se le puede cambiar el valor dependiendo del criterio que se tenga en ese momento. Hay que tener encuenta que al poner el cero significa que el número de solicitudes es indefinida.

Uno de los criterios en que podríamos basarnos es: Saber el número de objetos por cada página y el número mayor ponerlo en la directiva. Y de esta manera estaríamos optimizando de acuerdo a lo que tenemos.

10. Habíamos dicho que una conexión puede terminar por dos razones: (1). Por que las peticiones hayan terminado y (2). Por que el tiempo haya espirado. Hasta aquí hemos configurado el número de peticiones que debe tener una conexión. Esto quiere decir que si configuramos a nuestro servidor de la manera como está actualmente la conexión terminaría luego que el cliente termine de realizar todas las peticiones.

    Ilustremos esto mejor con un ejemplo: hemos configurado para que el servidor web atienda a 100 peticiones del cliente. Pero que pasa si las peticiones del cliente solo llegan a 30 y no esta configurado el tiempo de expiración. En este caso lo que ocurre con el cliente y el servidor es una conversación que llamaremos "*Conversación Pobre*" por que la información que viaja en esos paquetes no tiene mayor importancia, pero si tiene razón de ser, la cual es que el cliente y el servidor deben de cumplir con las 100 peticiones para poder cerrar la conexión.

    Entonces con lo anteriormente dicho podemos darle a nuestro servidor web las dos opciones para cerrar una conexión que es cerrar a través de un tiempo de expiración. Esta configuración la hacemos mediante la directiva *KeepAliveTimeout* en la cual se especifica el número de segundos que van a transcurrir antes de cerrar la conexión. Un ejemplo de cómo se configura esta directiva es el siguiente:

    KeepAliveTimeout 15

    Cuando configuramos a nuestro servidor para que cierre por las dos formas no quiere decir que va ha desaparecer la "*Conversación Pobre*", pero sí dependiendo del tiempo configurado puede disminuir, quedar igual o aumentar.

    Es por esto que para mejorar el rendimiento es necesario establecer un balance entre el número de peticiones y el tiempo de expiración de la conexión. Pero este balance no es el mismo para todos los servidores web, ya que no todos tienen las

mismas necesidades y concurrencia, en otras palabras no todos los servidores web son iguales.

Para realizar este balance hay que tener encuenta los siguientes aspectos:

- Cual es la página con mayor objeto, y cuantos objetos son.
- De todos los objetos que se almacenan en el servidor web cales son los más pesados y cuanto tardan en cargarse, aquí hay que tener en cuanta que no por que los objetos tarden 5min. En cargarse hay que tener un tiempo de expiración de 5 min. Además porque no es ese el requisito único y necesario para configurar el tiempo.
- La capacidad que tiene nuestro servidor web para servir las peticiones.
- Como esta configurado nuestra red interna para el control de tráfico y de ancho de banda.
- Hay que tener encuenta que no todos los clientes que nos visitan son clientes tienen un buen rendimiento en términos de tiempo del procesador. Este punto parece ser no tan importante pero si no lo tomamos encuenta puede suceder y de hecho va ha suceder que muchas personas no puedan acceder a nuestras páginas porque sus PCs (clientes) son lentas en tiempo de procesamiento.

Entonces que podemos hacer: verificar estos puntos especialmente los cuatro primeros, porque son los que podemos comprobar con nuestro servidor web, el quinto punto debemos haciendo, por ejemplo, una estimación tomando una PC (cliente) con esas características y calcular el tiempo.

11. Una directiva muy importante al momento que se construya URL de redireccionamiento, ósea cuando esta directiva se establece en on hará que Apache utilice los valores de las directivas *ServerName* y *Port*, y cuando se establece en off el servidor seguirá usando lo que le pase el cliente.

    UseCanonicalName Off

12. Lo siguiente que se debe determinar es que es lo que se va abrir cuando un URL solicita un directorio.

Por ejemplo, si el URL es: *http://www.espoch.edu.ec/servicio/*, el servidor consultará el *DirectoryIndex* (que es el que determina el tipo de archivo que va a ser mostrado) para ver si se encuentra ese directorio y servir ese archivo hacia fuera del mismo. Si se establece index.html en el *DirectoryIndex*, esta solicitud que daría: *http://www.espoch.edu.ec/servicio/index.html*.

DirectoryIndex index.html index.htm /var/

13. hagamos un recuento de lo ya estudiado anteriormente. En los registros del servidor (registros log) se registran todas las operaciones que se desprenden a raíz del llamado de una página web.
Cuando acceden a nuestro servidor web, ese acceso se almacena en el *Registro de Acceso* también llamado *Access log*,

En este registro se guarda entre otras cosas la identificación del host que ingresa a nuestro servidor.

Es precisamente esa información el tema a tratar en este punto. Para que el servidor pueda registra en el *Access Log* el nombre del host que realiza la petición debe de estar configurado para que pueda realizar esa función.
Esa configuración se la realiza con la directiva *HostNameLookups* en *on*. Pero que ocurre con esta configuración. Cuando se realiza esta configuración el servidor tiene que realizar DNS en el acceso de cada cliente para obtener el nombre del host. Dado que esto requiere que se realice una búsqueda DNS para cada documento que solicite el cliente, esto podría ralentizar todo significativamente.

Una recomendación que hacemos con referencia a esto es que no se debe poner en on a menos que halla una buena razón para hacerlo.

Esto quiere decir que si configuramos la directiva *HostNameLookups* en *off*, mejora el rendimiento de nuestro servidor, así como también los tiempos de respuesta, ya que el nombre del host pasará a los registros y se pasará a CGI en variables de entornos REMOTE_HOST como una dirección IP.

También se puede realizar una búsqueda inversa para por cuestiones de seguridad. Esto se lo realiza poniendo *HostNameLookups* en *double*, lo que

significa, que cuando se busca un nombre, se hace una búsqueda en ese nombre y la dirección IP resultante debe coincidir con la dirección ip del cliente.

14. En ocasiones cuando entramos en Internet y solicitamos una página y resulta que la dirección que pusimos en el buscador yo no es al correcta, el servidor de la página que vamos a buscar nos devuelve una página de error que indica que no a sido encontrada dicha página, incluida a esa información también aparece otra información adicional que es la información del servidor como por ejemplo la versión y el tipo del mismo.

    Siguiendo con el ejemplo, a nosotros como usuarios de esa página no nos interesa aquella información (referente al servidor), ya que lo que nos interesa en realidad es que aquel servidor no entregue la página que hemos solicitado.

    Entonces, adonde queremos llegar con esta información es que en la configuración de nuestro servidor hay que quitar esa información adicional ya que a ninguna de las personas que desean visitar nuestras páginas necesita saber acerca de nuestro servido.

    En lugar de ello mostrar más información de cómo poder arreglar ese error (dependiendo de cual sea) y poder ingresar a la página solicitada.

    Para poder mejorar esto que no es un error sino que es ocultar información que no necesita estar mostrada lo podemos hacer de las siguientes formas:

    → Para sacar la información de las páginas de error de nuestro servidor lo hacemos con la siguiente directiva:

    ServerSignature off

    No solamente en las páginas de error aparece información acerca de nuestro servidor, sino también en las cabeceras de los paquetes http que envía nuestro servidor a los clientes, y para sacar esa información lo hacemos con la siguiente instrucción

    → Para sacar la información de las cabeceras de respuesta http de nuestro

servidor lo hacemos con la directiva:

ServerTokens SO

17. Hay personas que cuando están navegando en Internet dejan un momento sin hacer algo, es decir, por un momento no hacen nada, y la página que ellos han pedido se encuentra visualizada en su navegador. Cuando esto ocurre el servidor web califica a los procesos producidos por la petición de esta página como procesos inactivos. Sabemos que todo proceso producido en un computador consumen tiempo del procesador y otros recursos como memoria virtual cuando se encuentra en la cola de espera, esperando tiempo del procesador.
    Esto quiere decir que tener varios procesos inactivos en un servidor web con una concurrencia considerable (con un alto número de visitas) perjudica al rendimiento del mismo.

    Esto que comienza como algo inocente se convierte en un problema. Este problema se puede solucionar por medio de las siguientes directivas:

    MinSpareServers 16
    MaxSpareServers 30

    La primera determina el mínimo de procesos inactivos y la segunda el máximo de procesos inactivos, de esta manera podemos controlar que el rendimiento de nuestro servidor no se vea afectado por estos tipos de procesos.
    Es muy importante tener encuenta que los procesos inactivos se alojan en memoria, por lo que declarar una cantidad mínima y máxima de estos procesos puede afectar la capacidad de memoria RAM del equipo. Por lo que se recomienda mucha RAM y una muy buena CPU.
18. Una opción importante y peligrosa a la vez, es la de especificar el número de solicitudes que va a servir un solo proceso secundario antes de que se cierre el proceso secundario. Lo peligroso de esta opción es que cuando se establece en cero el proceso secundario nunca se cierra, causando una baja del rendimiento cuando otros procesos se cierran. Esta opción podemos configurarla de la siguiente manera.

    MaxRequestsPerChild 30

19. Como ya hemos determinado cuantas son las solicitudes que se van a servir en un proceso secundario, tenemos ahora que establecer el número de solicitudes simultaneas del cliente que van a ser atendidas, esto lo hacemos:

    MaxClients 100

20. También debemos establecer un número máximo de solicitudes que pueden estar en la cola para ser atendidos. Hay que resaltar que esto tiene repercusión en sitios de mucho tráfico.

    ListenBacklog 511

    El número 511 es el recomendado.

21. Otra cosa que tenemos que tener en cuenta es la seguridad en las conexiones a nuestro servidor web. Ofrecerle a los clientes la posibilidad de elegir entre una conexión en un entorno seguro o no hace que el servidor sea más aceptable.

    Una de las formas de proporcionar seguridad es a través de un sistema de seguridad con certificador digitales que no permita ofrecer conexiones a nuestro servidor web en forma encodificado y seguro. Esto lo podemos realizar con el modulo ssl de apache (mod_ssl) que da soporte al "Secure sockets layer" (ssl) y "Transport layer security" (tls) entre un servidor web y los clientes.

    Como hacerlo.

    → Hay que tener en cuente lo siguiente: si estamos trabajando con Linux Red Hat 9 y hemos instalado el sistema operativo con Apache, este ya está listo con el modulo ssl. Pero si no ha instalado el Apache que viene con el Red Hat 9, entonces debe utilizar un paquete RPM que se llama: mod_ssl-2.0.xxxx.rpm.

    Este paquete rpm genera e instala lo siguiente:

    Certificador digitales locales en /etc/httpd/conf.
    El modulo por mod_ssl en /etc/httpd/modules.
    Archivo de configuración /etc/httpd/conf.d/ssl.conf

→ Una vez instalado el modulo mod_ssl de Apache por cualquiera de las dos vías, lo que si que sigue es dar una mirada a los siguientes archivos para revisar su contenido, estos archivos son los siguientes:

/etc/httpd/conf/ssl.crt/server.crt
/etc/httpd/conf/ssl.key/server.key

→ Ahora vamos a generar un certificado. Lo primero que debemos hacer es sacar un respaldo de los archivos de configuración de Apache por si llega a ocurrir algo.

mkdir /tmp/apache.
cp –r /etc/httpd/conf/ /tmp/apache/.*

→ Luego hacemos los siguientes pasos:

mkdir /etc/httpd/conf/tmp
cd /etc/httpd/conf/tmp
Primero
[root@localhost tmp]# *openssl genrsa -des3 -out server.key 2048*
Generating RSA private key, 1024 bit long modulus
.....++++++
.....++++++
e is 65537 (0x10001)
Enter pass phrase for server.key: [contraseña va aqui]
Verifying - Enter pass phrase for server.key: [contraseña va aqui]
[root@localhost tmp]#
Segundo
[root@localhost tmp]# *openssl rsa -in server.key -out server.pem*
Enter pass phrase for server.key: [contraseña de antes va aqui]
writing RSA key
[root@localhost tmp]#
Tercero
[root@localhost tmp]# *openssl req -new -key server.key -out server.csr*
Enter pass phrase for server.key: [contraseña de antes va aqui]
You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [GB]:EC

State or Province Name (full name) [Berkshire]:Riobamba

Locality Name (eg, city) [Newbury]:Norte

Organization Name (eg, company) [My Company Ltd]:Tesis

Organizational Unit Name (eg, section) []:ESPOCH

Common Name (eg, your name or your server's hostname) []:Escuela del Chimborazo

Email Address []:root@localhost

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []: [blanco]

An optional company name []: [blanco]

[root@localhost tmp]#

Cuarto

[root@localhost tmp]# *openssl x509 -req -days 60 -in server.csr -signkey server.key -out server.crt*

Signature ok

subject=/C=EC/ST=Quito/L=Sangoquil/O=CEDIA/OU=ESPE/CN=Escuela de Ejercito/emailAddress=root@localhost

Getting Private key

Enter pass phrase for server.key: [contraseña de antes va aqui]

[root@localhost tmp]#

→ Lo próximo a realizar es: vamos a /etc/httpd/conf y hacemos

cd /etc/httpd/conf

cp tmp/server.crt ssl.crt/.

cp tmp/server.key ssl.key/.

cp tmp/server.csr ssl.csr/.

service httpd stop

service httpd start

→ Al final de esta configuración el servidor Apache va a pedir una contraseña, entonces ingresamos la misma contraseña que pusimos en la configuración.

→ Ahora damos una mirada en /var/log/messages. Si al momento de la configuración dio un problema, probablemente un mensaje acerca de el error aparecerá en /var/log/messages

→ Ahora tratemos de abrir la página con la siguiente instrucción:

https://localhost/

→ Probablemente pase lo siguiente: Ahora apache pide una contraseña para inicializar. Desafortunadamente esto no va a funcionar en un ambiente de un servidor web.
Entonces lo que tenemos que hacer es sacar la contraseña y esto lo hacemos usando el archivo server.pem, este archivo es igual a server.key pero no está encodificado. Para hacer esto haz lo siguiente en /etc/httpd/conf:

cp tmp/server.pem ssl.key/server.key

→ Ahora reinicia el servidor de Apache con la siguiente instrucción:

Service httpd restart

Enjaulamiento Web

El siguiente paso es limitar a los procesos de Apache a que accedan a los sistemas de ficheros. La técnica del enjaulado, es como se describe a continuación:

Código:

```
mkdir -p /chroot/httpd/dev
mkdir -p /chroot/httpd/etc
mkdir -p /chroot/httpd/var/run
mkdir -p /chroot/httpd/usr/lib
mkdir –p /chroot/httpd/usr/libexec
mkdir –p /chroot/httpd/usr/local/apache2/bin
mkdir -p /chroot/httpd/usr/local/apache2/lib
mkdir -p /chroot/httpd/usr/local/apache2/logs/www.ebank.lab
```

mkdir -p /chroot/httpd/usr/local/apache2/logs/www.test.lab

mkdir -p /chroot/httpd/usr/local/apache2/conf

mkdir -p /chroot/httpd/usr/local/lib

mkdir -p /chroot/httpd/www

El dueño de los directorios anteriores debe ser el usuario root, y los derechos de acceso no deberían permitir que usuarios normales hagan cambios en dichos directorios:

Código:

chown -R root:sys /chroot/httpd

chmod -R 0755 /chroot/httpd

Ahora crearemos, si no lo está ya, el dispositivo especial, /dev/null.

Código:

ls -al /dev/null
crw-rw-rw- 1 root wheel 2, 2 Mar 14 12:53 /dev/null
mknod /chroot/httpd/dev/null c 2 2
chown root:sys /chroot/httpd/dev/null
chmod 666 /chroot/httpd/dev/null

También necesitamos crear un dispositivo /chroot/httpd/dev/log que será necesario para que el servidor trabaje correctamente. En el caso de nuestro sistema FreeBSD, la siguiente línea debería añadirse a /etc/rc.conf:

Código:

syslogd_flags="-l /chroot/httpd/dev/log"

Para que los cambios surtan efecto, necesitamos también reiniciar el demonio syslogd con el nuevo parámetro:

Código:

kill `cat /var/run/syslog.pid`

/usr/sbin/syslogd -ss -l /chroot/httpd/dev/log

El siguiente paso es copiar todos los programas, librerías y archivos de configuración necesarios en el nuevo árbol de directorios. Para el caso de FreeBSD 4.1 la lista de los archivos requeridos es la siguiente:

Código:

```
cp /usr/local/apache2/bin/httpd /chroot/httpd/usr/local/apache2/bin/
cp /usr/local/apache2/lib/libaprutil-0.so.9 /chroot/httpd/usr/local/apache2/lib/
cp /usr/local/apache2/lib/libapr-0.so.9 /chroot/httpd/usr/local/apache2/lib/
cp /usr/local/apache2/conf/mime.types /chroot/httpd/usr/local/apache2/conf/
cp /usr/local/apache2/conf/httpd.conf /chroot/httpd/usr/local/apache2/conf/
cp /usr/local/lib/libexpat.so.4 /chroot/httpd/usr/local/lib/
cp /usr/lib/libc.so.5 /chroot/httpd/usr/lib/
cp /usr/lib/libcrypt.so.2 /chroot/httpd/usr/lib/
cp /usr/lib/libm.so.2 /chroot/httpd/usr/lib/
cp /usr/libexec/ld-elf.so.1 /chroot/httpd/usr/libexec/
cp /var/run/ld-elf.so.hints /chroot/httpd/var/run/
cp /etc/hosts /chroot/httpd/etc/
cp /etc/nsswitch.conf /chroot/httpd/etc/
cp /etc/resolv.conf /chroot/httpd/etc/
cp /etc/group /chroot/httpd/etc/
cp /etc/master.passwd /chroot/httpd/etc/passwords
```

Para el caso de otros sistemas Linux, Unix y BSD, la lista de los archivos necesarios puede determinarse usando los comandos ldd, strace, truss o strings.
Después de que los pasos anteriores están completados, necesitamos preparar la base de datos de passwords que debe haber en el presente sistema enjaulado. Así, de /chroot/httpd/etc/passwords y /chroot/httpd/etc/group tenemos que quitar todas las líneas, excepto apache. Ahora, deberíamos construir la base de datos de passwords como sigue:

Código:

```
cd /chroot/httpd/etc
pwd_mkdb -d /chroot/httpd/etc passwords
rm -rf /chroot/httpd/etc/master.passwd
```

Estos comandos deben ejecutarse si se usa FreeBSD. En otros sistemas, podría ser suficiente con editar los archivos /chroot/httpd/etc/passwd y /chroot/httpd/etc/shadow.

Finalmente, podemos copiar el contenido de ejemplo del sitio web al entorno enjaulado:

Código:

```
cp -R /www/* /chroot/httpd/www/
```

Y comprobar que el servidor Apache funciona correctamente:

Código:

```
chroot /chroot/httpd /usr/local/apache2/bin/httpd
```

Pasos finales.

Si tu Apache ahora funciona correctamente, lo único que nos queda es crear un script que ejecute Apache en tiempo de arranque. Para hacer esto, puede utilizarse el script apache.sh:

Código:

```
#!/bin/sh
CHROOT=/chroot/httpd
HTTPD=/usr/local/apache2/bin/httpd
PIDFILE=/usr/local/apache2/logs/httpd.pid

echo -n " apache"

case "$1" in
start)
    /usr/sbin/chroot $CHROOT $HTTPD
    ;;
stop)
    kill `cat ${CHROOT}/${PIDFILE}`
    ;;
*)
    echo ""
    echo "Usage: `basename $0` {start|stop}" >&2
    exit 64
    ;;
esac

exit 0
```

El anterior script debería copiarse al directorio donde los scripts de arranque están ubicados. Para el caso de FreeBSD es /usr/local/etc/rc.d. Los derechos de acceso que el fichero debe contener deberían ser:

Código:

```
chown root:sys /usr/local/etc/rc.d/apache.sh
chmod 711 /usr/local/etc/rc.d/apache.sh
```

4.1.1.3 Aceptabilidad

La aceptabilidad de un servidor Web, puede ser medida por varios aspectos, entre los cuales se encuentra uno muy importante, que es quienes van a decidir si el servidor es aceptable o no. En esta metodología se han definido tres parámetros para decir que un servidor Web Apache es aceptable, en base a los siguientes aspectos:

Funcionamiento:

- Ejecutar Apaci y sus comandos para la configuración de Apache como:
  -Show Layout.
  -Prefix
  -Enable-Module.

Rendimiento y Seguridad.

- Configuración del archivo httpd.conf con las siguientes directivas:
  - Server Root Establece el directorio en el cual se van a guardar los archivos del servidor. Esta directiva debe estar en Sever Root "etc/httpd"
  - PidFile Específica la ubicación del número del PID del servidor. Esta directiva debe estar en Pid File run/httpd.pid.
  - Listen Indica al sevidor que escuche las solicitudes de más de una dirección IP y/o puerto TCP/IP. Esta directiva debe estar en Listen 127.0.0.1:80
  - Document Root Directorio desde el cual se van a copiar los archivos html.Esta directiva debe estar en Document Root"/var/www/html/".
  - ServerAdmin Dirección de correo de la persona responsable del servidor.
  - ServerName Se especifica el nombre del servidor como por ejemplo:www.espoch.edu.ec.
  - TimeOut para medir el tiempo de duración entre el cliente y el servidor. De acuerdo al criterio del administrador.
  - keepAlive Sirve para hacer varias solicitudes en una misma conexión.
  - MaxKeepAliveRequest cuantas son las solicitudes que se van a servir por dicha conexión.
  - KeepAlive Para mejorar problemas de rendimiento cuando el servidor tiene muchas visitas. Esta directiva debe estar en ON.

- MaxKeepAliveRequest Peticiones que hace el servidor por cada conexión. **A criterio del Administrador.**
- Keep Alive Timeout # de segundos que van a transcurrir antes de cerrar la conexión. **A criterio del Administrador.**
- Use Canonical Name.
- Directory Index Determina el tipo de archivo, que va ser mostrado. Esta directiva debe estar así: index.html index.html/var/.
- AccessLog Registro de acceso a logs
- HostnameLooKups para que pueda registrar los logs. Esta directiva debe de estar así: Hostname lookups off.
- Server Signature Sacar información de las paginas de error de nuestro servidor lo hacemos con la siguiente directiva: Server Signatura off.
- Server Tokens Para sacar información de las cabeceras de respuesta http de nuestro servidor. La directiva debe ser de la siguiente manera Server Tokens ponerlo **a criterio del administrador.**
- MinSpareServers Máximo de procesos inactivos. Esta directiva debe estar de la siguiente manera: MinSpareServers 16 por defecto esto queda a **criterio del administrador.**
- MaxSpareServers Máximo de procesos inactivos. Esta directiva debe estar a la siguiente manera MaxSpareServers 30 por defecto esto queda a **criterio del administrador**.
- MaxClients Número de solicitudes simultáneas del cliente .Esta directiva debe estar de la siguiente manera: MaxClients 100 esto queda **a criterio del administrador.**
- Listen Backlog Número máximo de solicitudes que pueden estar en la cola para ser atendidas. Esta directiva debe estar de la siguiente manera: Listen Backlog 511.

Cumplidas todas estas configuraciones en nuestro servidor Web, se ha a podido mejorar su funcionamiento, rendimiento y seguridad. Al haberse cumplido todos los parámetros podemos decir que nuestro servidor Web es aceptable.

4.2. IIS (Internet Information Server)

Esta metodología esta basada en los siguientes parámetros:

Funcionamiento

Rendimiento

Seguridad

Aceptabilidad.

4.2.1 Funcionamiento de Internet Information Server

Para empezar a explicar el funcionamiento de IIS 6.0, primero se explicara el modo de instalarlo.

Instalar IIS utilizando el Asistente para configurar el servidor

1. En el menú Inicio, hacer clic en Administre su servidor.

Figura 35 Grafica de opción Administre su servidor

2. En Administrar las funciones de su servidor, hacer clic en Agregar o quitar función.

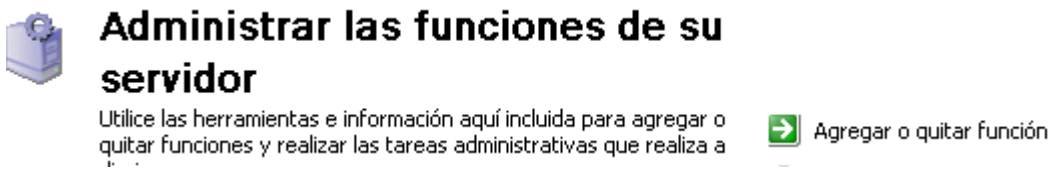

Figura 36 Opciones de agregar o quitar función

→ Leer los pasos preliminares indicados en el Asistente para configurar su servidor y hacer clic en Siguiente.

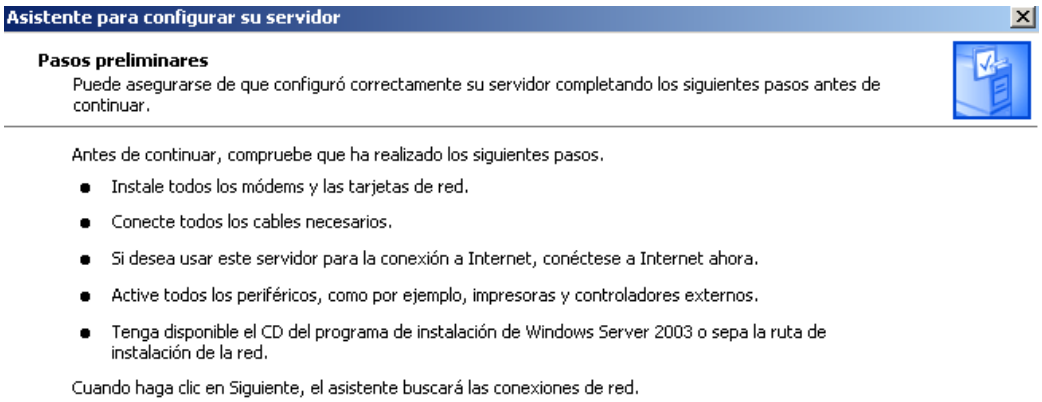

Figura 37 Asistente para configurar el Servidor

3. En Función del servidor**,** hacer clic en Servidor de aplicaciones (IIS, ASP.NET) y a continuación, en Siguiente**.**

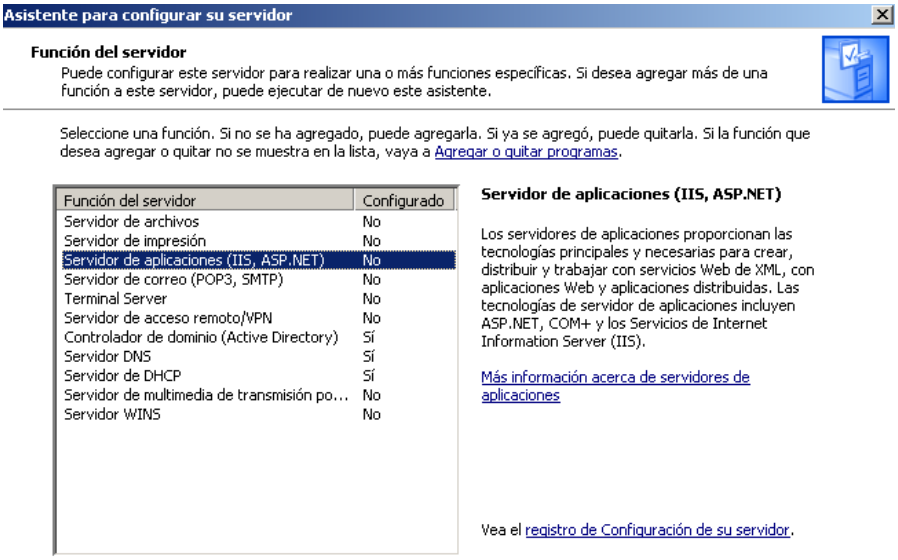

Figura38 Funciones del Servidor

4. Seleccionar las herramientas adicionales:

Extensiones de servidor de Front Page. Se utilizan para publicar con Front Page, Visual Studio y carpetas Web.

Habilitar ASP.NET. ASP es un entorno de programación para desarrollar aplicaciones y servicios basados en Web accesible para cualquier explorador o dispositivo.

Figura 39 Opciones del Servidor de aplicaciones

5. Hacer clic en Siguiente.

Leer el resumen de las selecciones y hacer clic en Siguiente.

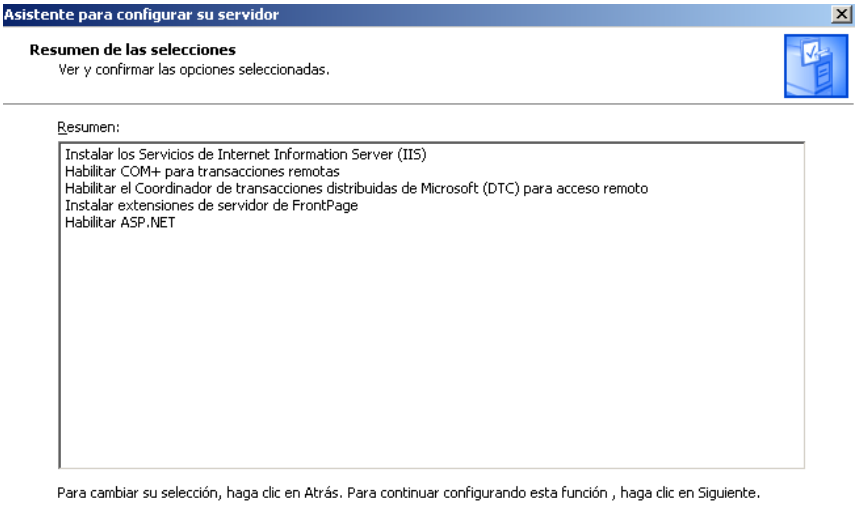

Figura 40 Resumen de Selecciones

6 Luego saldrá la pantalla que nos permite ver, la agregación de funciones que hemos escogido.

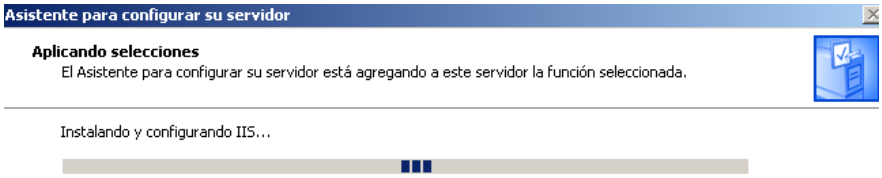

Figura 41 Aplicación de Selecciones

7 Después se pide que insertemos el disco de Windows 2003 para instalar estos componentes.

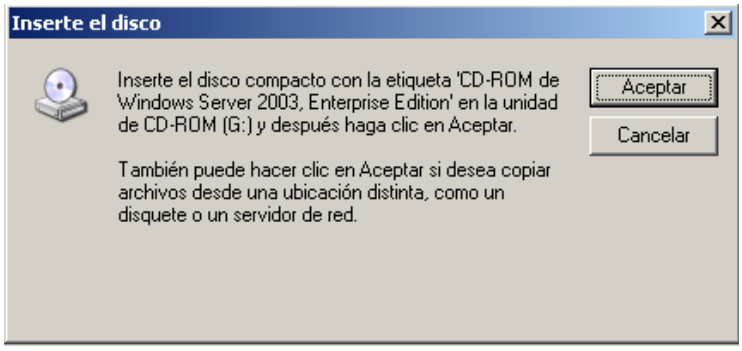

Figura 42 Pantalla para instalar aplicaciones seleccionadas

8 Se comienzan a instalar los componentes, como lo muestra la siguiente pantalla:

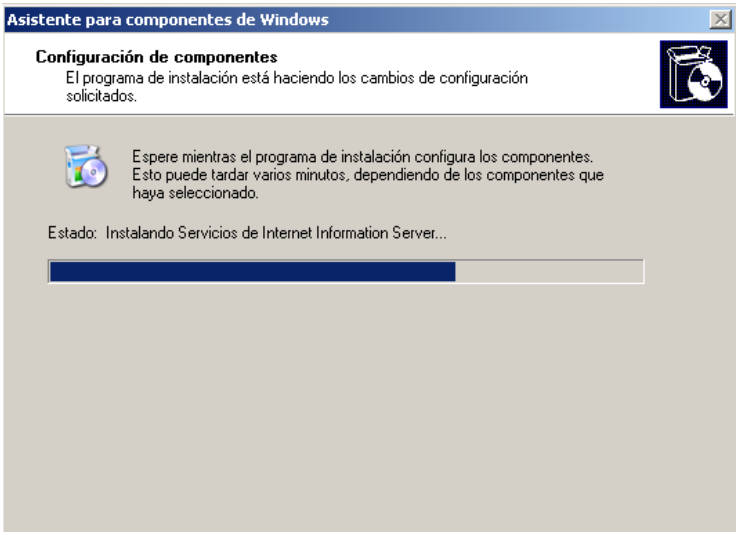

Figura 43 Configuración de Componentes

9. Al finalizar la instalación de los componentes, se nos muestra una pantalla diciendo que se ha instalado correctamente Internet Information Server, para terminar hacer clic en Finalizar.

Figura 44 Pantalla de Finalización de la Instalación

Para comprobar si el servidor de IIS funciona correctamente, utilizar un explorador de Internet para ver la página Web. Intentar ver la página Web en un equipo que esté en la misma red de área local (LAN) que el servidor de IIS que se está probando. Para comprobar esto nos vamos a Inicio, Herramientas administrativas, Administrador de Internet Information Server, Sitios Web, Nuevo, Sitio Web, esto lo hacemos para que los equipos que estén en la misma red pueden ver la página y verificar si el Internet Information Server, ha sido instalado correctamente.

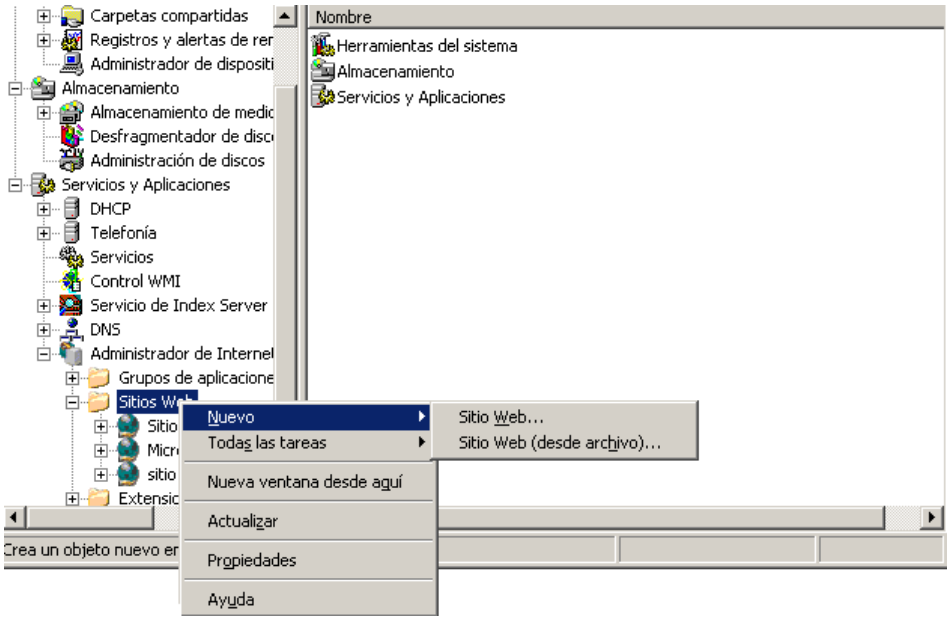

Figura 45 Creación de Sitio Web

Para comprobar si el sitio Web se encuentra disponible en Internet, examinarlo mediante un equipo que no esté en la misma red de área local (LAN) que el servidor de IIS que se está probando.

Una vez instalado Internet Information Server procedemos ha habilitar algunos de sus servicios.

4.2.2 Configuración

10. Una vez que tengamos instalado IIS, para ejecutar el administrador de Servicios de Internet Information Server, lo podemos hacer de varias formas:

- MMC (Microsoft Management Console).- Para esta consola se da clic derecho sobre mi PC y se da clic en administrar y sale la siguiente pantalla.

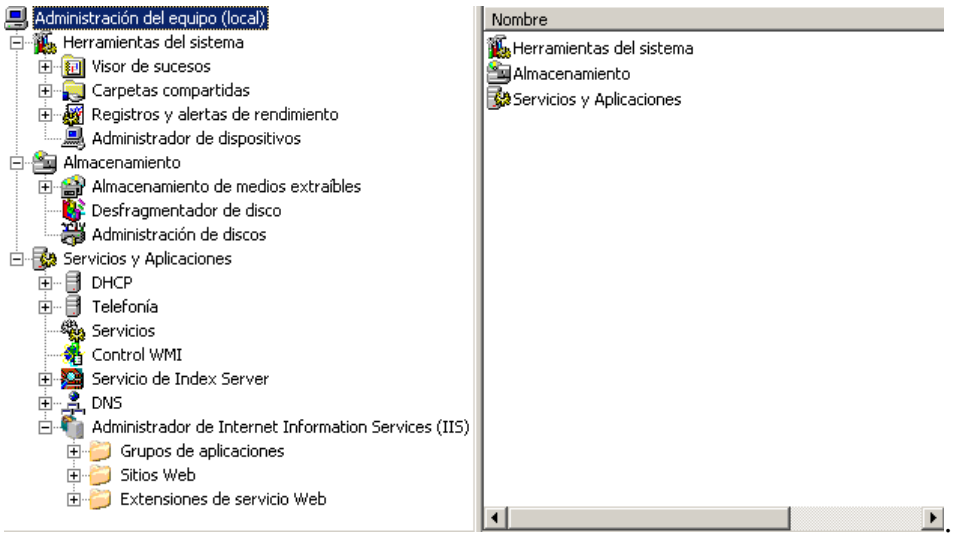

Figura 46 Administrador del IIS desde mi PC

- Herramientas administrativas: Servicios de Internet Information Server. Esta opción se va por inicio herramientas administrativas, administrador de Internet Information Server.

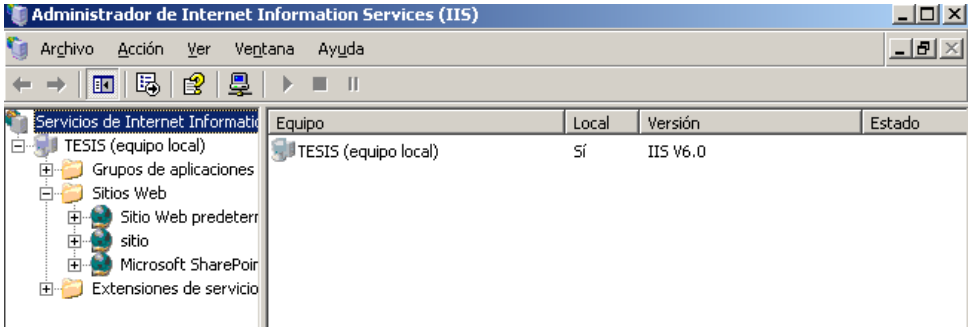

Figura 47 Administrador del IIS desde Herramientas Administrativas

11. Ejecutando InetMGR.exe: lo podemos encontrar en systemroot%/system32/inetsrv**.** Al dar click en este archivo aparece la misma pantalla que al seleccionar herramientas administrativas.

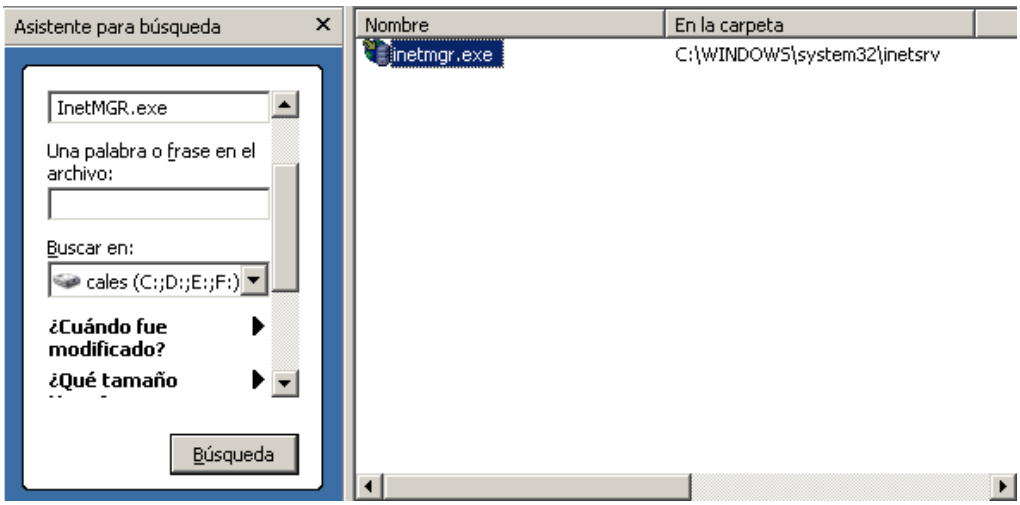

Figura 48 Busqueda de archivo para mostrar Administrador de IIS

Opciones de Configuración

Veremos las opciones de configuración por medio de MMC, también puedes utilizar el explorador para realizar las configuraciones, esto se realizaría por el sitio Web de administración (esto se usa fundamentalmente para hacerlo remotamente)

Propiedades de los sitios web

Sitio Web

Esta hoja de propiedades se utiliza para configurar básicamente el sitio web:

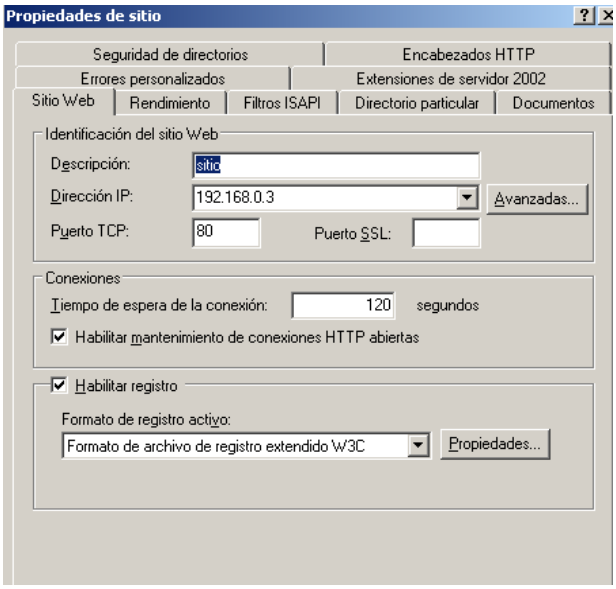

Figura 49 Propiedades del Sitio Web

12. Estas opciones salen por defecto en el sitio, se las deja tal y como muestra la pantalla. En la opción formato de archivo de registro: podemos escoger entre los tres formatos de archivos de registro, para ver cada una de sus propiedades damos clic en propiedades, nos aparecerá una pantalla cuya segunda ficha será como esta:

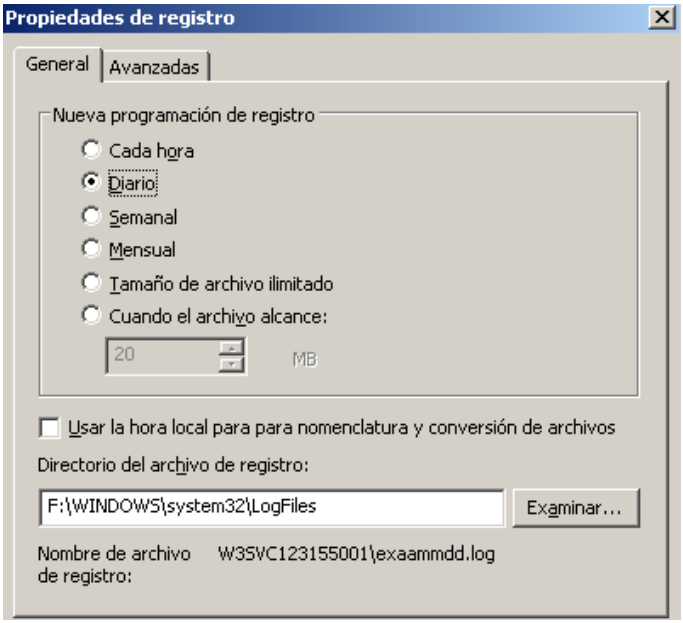

Figura 50 Propiedades del Registro

Esta pantalla permite indicar cada que tiempo deseamos que el registro se active, es recomendado que el archivo se registre diario para poder diariamente las peticiones que efectúa el servidor.

Formato de archivo de registro extendido W3C

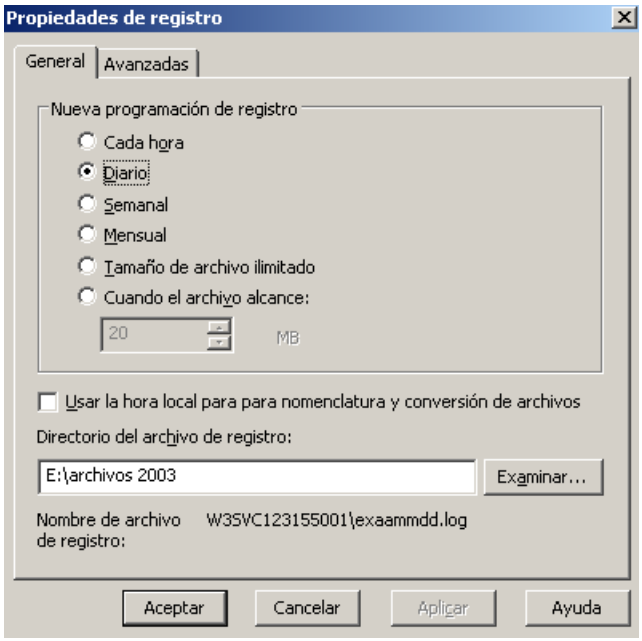

Figura 51 Propiedades del registro W3C

Formato de archivo de registro extendido W3C	Por el tamaño del archivo	extendnn.log
	Cada hora	exaammddhh.log
	Diario	exaammdd.log
	Semanal	exaammss.log
	Mensual	exaamm.log

Tabla 11 Formato de archivo de Registro extendido W3C

Este registro es el único que nos permite hacer configuraciones de cómo queremos que se vea nuestro archivo log.

Mensajes de error personalizados

13. Estos son algunos de los errores que se pueden personalizar y que son los más comunes:

Código de error	**Mensaje**
400	Solicitud incorrecta
403.1	Acceso de ejecución prohibido
403.2	Acceso de lectura prohibido
403.3	Acceso de escritura prohibido
403.8	Acceso al sitio denegado
403.14	Lista de directorios denegada

404	No se encuentra
404.1	Sitio no encontrado
500	Error interno del servidor
500-100.asp	Error ASP

Tabla 12 Codigos de error y sus mensajes

Los mensajes de error se muestran en una lista del complemento IIS que IIS trata como una sola propiedad. Por ejemplo, cuando se configura un conjunto de mensajes de error personalizados para el sitio Web, todos los directorios de este servidor heredan la lista completa de mensajes personalizados. Es decir, no se combinan las dos listas de mensajes de error personalizados (para el servidor y para el directorio).

El error 404.1 sólo se produce en equipos con direcciones IP múltiples. Si se recibe una solicitud de cliente en una combinación de dirección y puerto IP determinada y la dirección IP no está configurada para la recepción en ese puerto específico, IIS devolverá el mensaje de error HTTP 404.1. Por ejemplo, si un equipo dispone de dos direcciones IP y solamente una de ellas está configurada para escuchar en el puerto 80, cualquier solicitud con puerto 80 que se reciba en la otra dirección IP hará que IIS devuelva el mensaje de error 404.1.

Podemos asignar mensajes de error personalizados a un archivo o a una dirección URL para esto utilizaremos la hoja de propiedades Errores personalizados del complemento IIS:
Para personalizar un mensaje de error mediante su asignación a un archivo:

1. Creamos un archivo que contenga su mensaje de error personalizado y colocamos el archivo en un directorio.
   Seleccionamos el error HTTP que desea cambiar.
2. Hacemos clic en el botón Modificar propiedades.
3. Seleccionamos Archivo en el cuadro Tipo de mensaje.
4. Escribimos la ruta de acceso y el nombre del archivo que apunta al mensaje de error personalizado o utilizamos el botón Examinar para localizar el archivo en el disco duro del equipo.
5. Hacemos clic en Aceptar.

Nota si es una URL seleccionaremos "Dirección URL" en el cuadro Tipo de mensaje.

Podemos modificar los ficheros con FrontPage pero ojo, si incluimos un gráfico debe ser con la forma http://servidor/images/logo.gif. De lo contrario incluirá un enlace a una ruta local c:\imagenes\logo.gif que no funcionará bien.

Propiedades: Directorio Particular

14**.** Esta hoja de propiedades se utiliza para configurar el directorio de publicación del sitio Web:

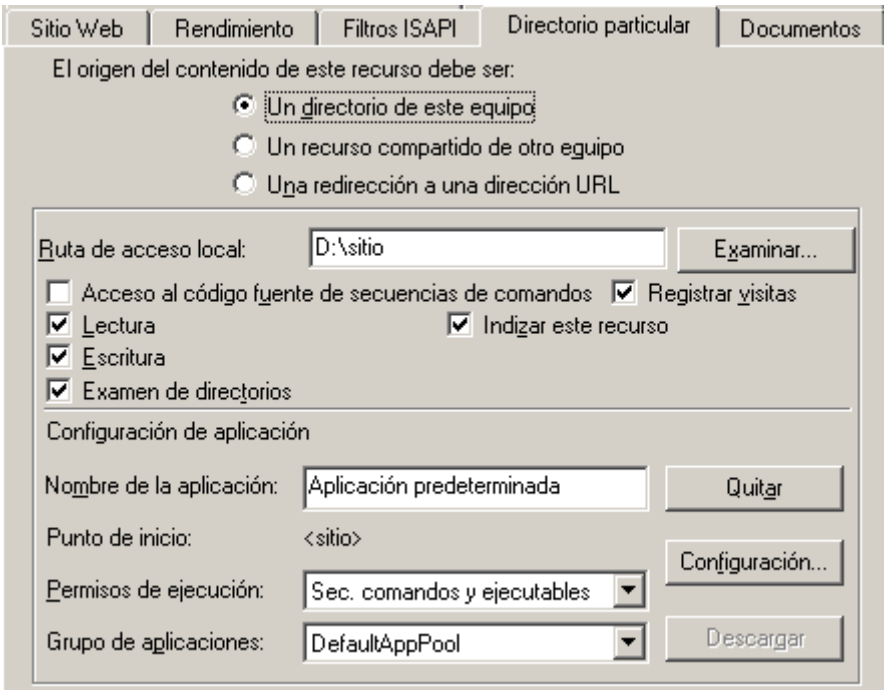

Figura 52 Propiedades del Directorio Particular

Origen del web

15. La primera parte hace referencia a la ubicación de las páginas web. Las tres opciones posibles son:

- Un directorio de este equipo: "c:\inetpub\wwwroot"
- Un recurso compartido de otro equipo. Si seleccionamos esta opción debemos indicar la ruta UNC \\servidor\recurso. Si además debemos identificarnos en ese directorio pulsaremos en el botón "conectar como" que se mostrará al elegir esta opción:
- Poco utilizada: redirección a una URL. Redirige un directorio virtual a la dirección URL de destino sin agregar ninguna parte de la dirección URL original. Puede utilizar esta opción para redirigir un directorio virtual completo a un archivo. Por ejemplo, para redirigir al archivo Predeter.htm en el directorio particular todas las peticiones realizadas al directorio virtual /scripts, escriba /Predeter.htm en el cuadro de texto Redirigir a y seleccione esta opción

Nota.-El administrador puede usar cualquiera de las dos primeras opciones dependiendo de su necesidad.

Permisos de ejecución

16. Existen varias formas de dar permiso de acceso al sitio web:

- Acceso al código fuente de secuencias de comandos. Selecciona esta opción para permitir que los usuarios tengan acceso al código fuente si disponen de los permisos de escritura o de lectura. El código fuente incluye las secuencias de comandos en aplicaciones ASP.
- Lectura. Esta opción permite que los usuarios lean o descarguen archivos o directorios y sus propiedades asociadas.
- Escritura. Permitir que los usuarios carguen archivos y sus propiedades asociadas en un directorio con este permiso del servidor o para cambiar el contenido de un archivo con este permiso. La escritura sólo se puede realizar con un explorador que admita la característica PUT del estándar de protocolo HTTP 1.1.
- Examinar directorios. Esta opción permite que el usuario vea una lista con formato de hipertexto de los archivos y subdirectorios de este directorio virtual. Los directorios virtuales no aparecen en las listas de directorios. Los usuarios deben conocer su alias. Importante El servidor Web presentará el mensaje de error "Acceso prohibido" en el explorador Web del usuario si éste intenta tener acceso a un archivo o directorio y las dos condiciones siguientes se cumplen:
    - La opción Examinar directorios está deshabilitada.
    - El usuario no especifica un nombre de archivo, como NombreArchivo.htm.
- Registrar visitas. Es opción registra las visitas de este directorio en un archivo de registro. Las visitas sólo se registran si está habilitado el registro para este sitio Web.
- Indizar este recurso. Selecciona esta opción para que Servicios de Microsoft Index Server incluya este directorio en un índice de texto del sitio Web.

Configuración de la aplicación

17. Esta es la parte que enlaza el mundo ASP y ASP.NET con el IIS. Una aplicación es la capacidad del IIS para ejecutar secuencias de comandos como ASP y ASP.NET. Es decir si esta sección está desactivada no funcionarán nuestra páginas ASP ni ASP.NET

Por lo tanto puede haber más de una aplicación por cada sitio Web. El sitio Web predeterminado creado al instalar Servicios de Internet Information Server es un punto de inicio de aplicaciones. Resumiendo, sólo necesitamos indicar un nombre para la aplicación y con esto IIS creará internamente la estructura necesaria para que pueda ejecutar páginas ASP entre otras.

Rendimiento de la aplicación

18. Hay un equilibrio entre el rendimiento y el grado de protección de la aplicación. Las aplicaciones que se ejecutan en procesos de servicios Web (Inetinfo.exe) obtienen un rendimiento superior, pero también existe un mayor riesgo de que una aplicación que no funciona correctamente pueda hacer que los servicios Web dejen de estar disponibles. La configuración recomendada es ejecutar inetinfo.exe en su propio proceso, ejecutar aplicaciones decisivas en sus propios procesos y ejecutar el resto de las aplicaciones en un proceso agrupado y compartido.

Observaciones:

- Para detener una aplicación y descargarla de la memoria, haz clic en el botón Descargar. Si el botón Descargar aparece atenuado, significa que no se encuentra en el directorio que sirve como punto de inicio de la aplicación.
- Si deseamos anular la asociación de este directorio principal con una aplicación, haga clic en el botón Quitar.
- Activaremos la casilla de verificación Ejecutar en otro espacio de memoria (proceso aislado) para ejecutar la aplicación en un proceso diferente del proceso del servidor Web. La ejecución de una aplicación aislada protege otras aplicaciones, incluido al propio servidor Web, de que se vean afectadas si esta aplicación tiene un error o deja de responder.

Establecer permisos para una aplicación

19. Los niveles de permisos para la aplicación son:

- Ninguno para impedir la ejecución de ningún programa o secuencia de comandos.
- Sólo secuencias de comandos para permitir la ejecución de las aplicaciones asignadas a un motor de secuencias de comandos en este directorio sin tener establecido el permiso Ejecución. Este es el que tenemos que tener activado para nuestras páginas ASP.NET. El permiso Secuencia de comandos es más seguro que el permiso Ejecución, ya que permite limitar las aplicaciones que se pueden ejecutar en el directorio.

- Establezca el permiso Secuencias de comandos y ejecutables para permitir la ejecución de cualquier aplicación en este directorio, incluso de las aplicaciones asignadas a motores de secuencias de comandos y archivos binarios de Windows (archivo .dll y .exe). (ojo con esto)

Establecer asignaciones para la aplicación

20. Para asignar una extensión a una aplicación:

En la consola administrativa seleccionamos el sitio Web o el directorio que sirve como punto de inicio de una aplicación.

1. Abrimos las hojas de propiedades del directorio y haga clic en la ficha Directorio particular, Directorio virtual o Directorio.
2. Hacemos clic en el botón Configuración.
3. En la ficha Asignaciones seleccionamos Agregar.
4. En el cuadro Ejecutable, escribimos la ruta de acceso del programa ISAPI o CGI que procesará el archivo. Debemos especificar un programa en un directorio local del servidor Web.
5. En el cuadro Extensión, escribimos la extensión de archivo que desea asociar al programa ISAPI o CGI. Cuando el servidor Web reciba una dirección URL que identifique un archivo con esta extensión, iniciará el programa asociado para procesar la petición.
6. Para permitir el procesamiento de archivos de este tipo en un directorio con el permiso Secuencia de comandos, activamos la casilla de verificación Motor de secuencias de comandos. Si un directorio tiene establecido el permiso Secuencia de comandos (en lugar del permiso Ejecución), sólo se podrán procesar en el directorio los archivos asociados a aplicaciones que sean motores de secuencias de comandos designados.

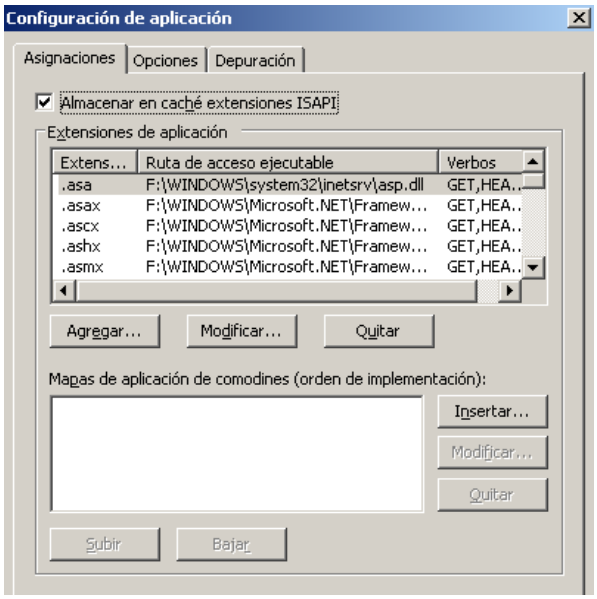

Figura 53 Configuracion de la Aplicación

Ficha de Opciones en propiedades de aplicación

21. Aquí vamos a comentar las opciones que nos aparecen en la pestaña Opciones, algunas de ellas muy importantes:

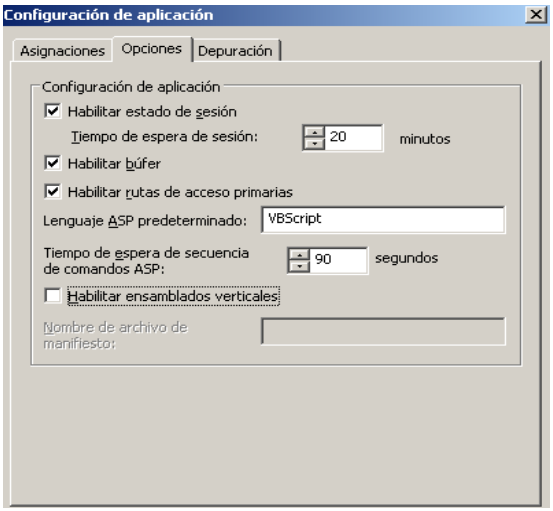

Figura 54 Opciones de la Configuración de la aplicación

- Habilitar estado de sesión. Esta opción es muy importante a la hora de trabajar con "variables globales" dentro de las ASP.
- Habilitar búfer. Esta opción está deshabilitada por defecto en IIS 4.0.
- Habilitar rutas de acceso primarias. Permite que las páginas ASP utilicen rutas de acceso relativas para el directorio primario del directorio actual (rutas de acceso con la sintaxis.)
- Lenguaje ASP predeterminado. Tenemos dos motores instalados: Vbscript y Javascript y podemos utilizar cualquiera de ellos. De forma predeterminada y el recomendado es VBScript

- Tiempo de espera: Especifica el intervalo de tiempo que ASP permitirá ejecutarse a una secuencia de comandos. Superado ese tiempo devolverá un error.

Ficha de Opciones en propiedades de aplicación

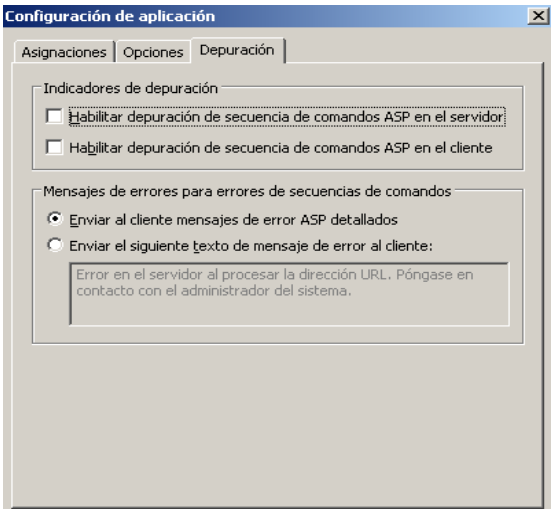

Figura 55 Ficha depuración de configuración de la aplicación

Permite operaciones como la depuración de las secuencias ASP. Tenemos dos niveles o tipos de depuración: en el cliente o en el servidor.

Propiedades: Documentos

Aquí establecemos los documentos que por defecto se servirá cuando se acceda al sitio Web. El orden que se establezca en dicha ventana será la que se utilice cuando se realice la petición. Es decir si tenemos default.html y default.asp, cuando en el explorador pongamos http://servidor/ se ejecutará como si hubiéramos tecleado http://servidor/default.html, en el caso que no existiera el fichero default.html se procedería a utilizar y/o buscar default.asp y así sucesivamente.

Habilitar el píe de página del documento, si queremos anexar automáticamente un píe de página con formato HTML en todos los documentos enviados por el servidor Web, utilizamos esta opción, debemos tener en cuenta que no debe de ser un documento HTML completo, solamente debemos incluir las etiquetas necesarias para definir la apariencia.

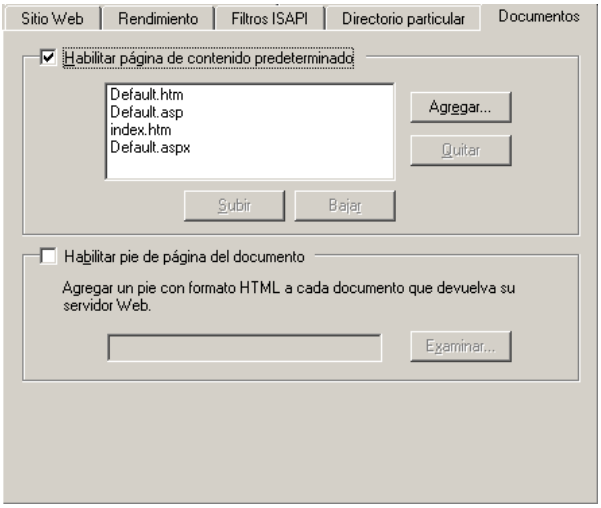

Figura 56 Ficha documentos en propiedades del sitio web

Encabezados HTTP

22. En esta hoja de propiedades vamos a poder establecer los valores de las cabeceras HTTP que se envíen a nuestros clientes (navegadores Web

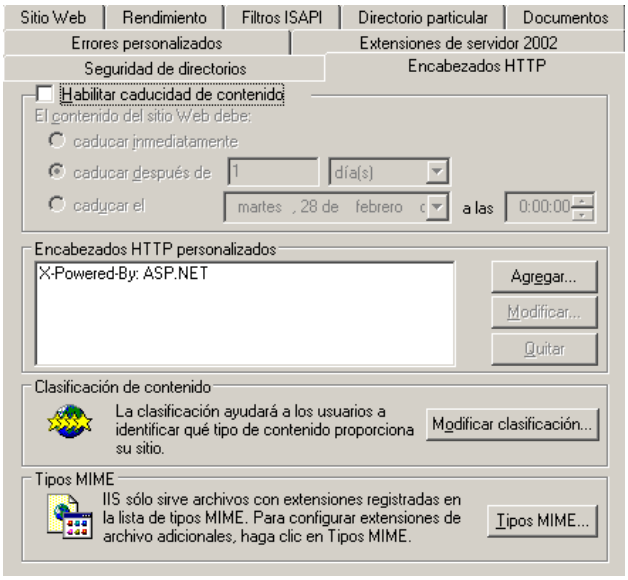

Figura 57 Encabezados HTTP

Si activamos la casilla de verificación Habilitar caducidad de contenido nos permitirá establecer el momento en el que la información de la página se considera no válida u obsoleta (ha caducado), el navegador Web comparará la fecha del sistema con la de la página que le llega y determinará si se visualiza la página almacenada en la caché o bien pedirá una actualización de la misma.

En la propiedad Encabezados HTTP personalizados podremos enviar un encabezado HTTP personalizado del servidor Web al navegador del equipo cliente.

Las páginas Web pueden incorporar cabeceras con información que identifique sus contenidos desde un punto de vista moral, la configuración de estas cabeceras la realizaremos a través del apartado "Restricción de contenido". De esta manera los usuarios pueden filtrar las páginas en función de ese contenido. IIS utiliza un método desarrollado por el organismo RSAC (Recreational Software Advisory Council) que clasifica los contenidos atendiendo a diversos grados de violencia, pornografía y lenguaje ofensivo. Para establecer estas restricciones de contenido se debe pulsar el botón "Modificar clasificación".

En la opción Tipos MIME (Multipurpose Internet Mail Extensions) podemos determinar los tipos de archivos que se enviarán al cliente Web.

Seguridad de directorios

23. Esta es una de las partes más complejas de la administración del sitio web con IIS. No es que revista dificultad sino que son los parámetros que van a determinar cómo va a funcionar nuestra Intranet. Además concederá y negará el acceso a usuarios y otros aspectos relativos a la seguridad.

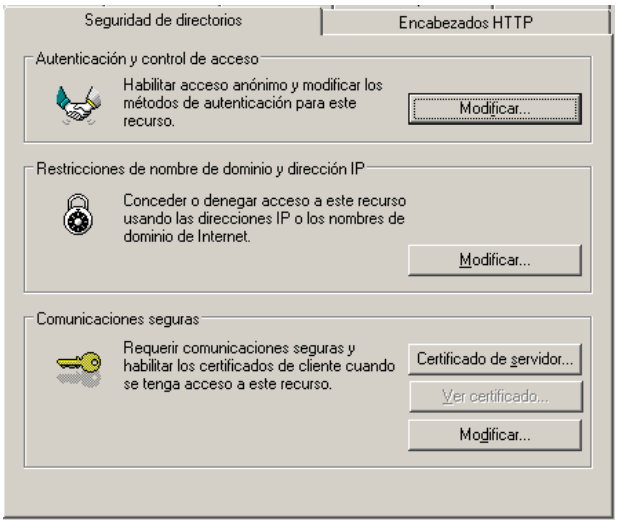

Figura 58 Seguridad de Directorios

Revisamos ya con la consola administrativa los niveles de seguridad que son:

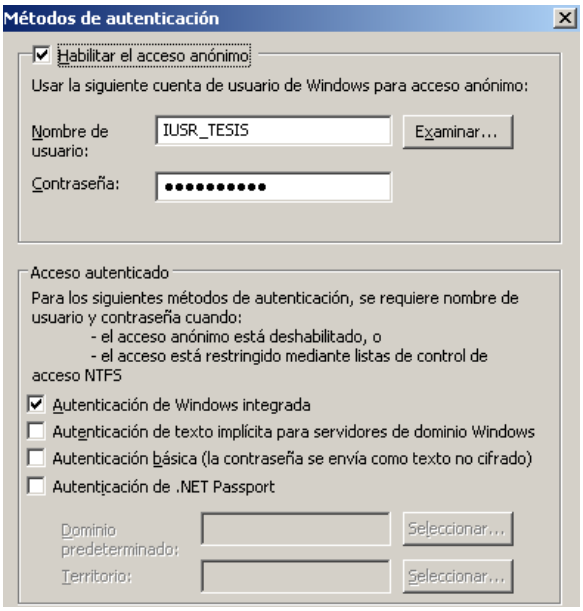

Figura 59 Metodos de Autenticación

Restricciones de nombres de dominio y direcciones IP

24. Otro de los métodos para controlar el acceso es a través de las direcciones IP. Si observamos la siguiente pantalla:

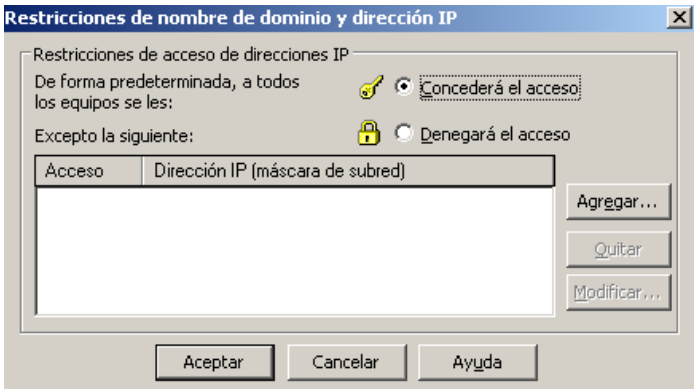

Figura 60 Restricciones de nombres de dominio y direcciones IP

Observamos que podemos indicar una dirección IP que será la o las excepciones de lo que indiquemos. Es decir, en el ejemplo por defecto se les deniega a todos el acceso excepto para la dirección 127.0.0.1, que es una dirección especial que indica al propio equipo.

Extensiones de servidor

25. Uno de los elementos mas importantes en nuestro servidor IIS son las "Extensiones de servidor". Esta es una aplicación incluida con IIS que permite conectar el programa FrontPage con el IIS. Por ejemplo desde FrontPage podremos crear y administrar distintos sitios Web trabajando con un IIS. Es decir le podemos decir "Crear un sitio Web" en el servidor "intranet". FrontPage buscará un servidor de

páginas Web (IIS) en esa ubicación y luego que esté el módulo de "extensiones de servidor". Si está correctamente podré recuperar los datos de este servidor: sitios web, permisos, y otros detalles que harán que no necesitemos manejar mas veces la consola administrativa de IIS.

Podemos ver que apenas hay opciones, la verdad es que no hay que activar prácticamente nada. Las opciones son:

- Habilitar edición (sólo para el web raíz). Permite que los autores utilicen FrontPage para acceder y modificar el web. Debe estar activada para que podamos editar el web con Frontpage
- Control de versiones. Si activamos esta opción podemos obtener información sobre quién está modificando el contenido del web, identificar los cambios o evitar que los cambios realizados por un autor borren los de otro. Podemos elegir el incorporado u otro externo mas potente como Visual Sourcesafe
- Rendimiento. Estimando el volumen de movimiento de nuestra web Frontpage reservará memoria para su caché.
- Secuencias de cliente. Cuando las extensiones de frontpage nos genere algún código podemos seleccionar el lenguaje.
- Especificar cómo se debe enviar el correo. Podemos especificar los parámetros se configuración de la cuenta y correo electrónico para los casos que se necesite.

El siguiente grupo de opciones utilizan las opciones del servidor raíz. Si queremos que no las hereden activaremos la casilla para conseguir:

- Registrar acciones de edición. El sistema graba quien ha realizado modificaciones: usuario, web, equipo remoto, ... estos datos se almacenan en un archivo: "_vti_log/Autor.log"
- Administrar permisos manualmente. Si está activada podemos utilizar las extensiones se servidor para administrar permisos, en caso contrario hay que realizar los cambios manualmente
- Solicitar SSL para edición. Permite autenticar a los autores de la web.

Por defecto los sitios Web incluyen las extensiones de servidor. Durante la vida de nuestro web es posible que éstas se deterioren: un corte de luz, corrupción interna,... entonces debemos comprobar y/o reinstalar estas extensiones. Para realizar esta operación debemos seleccionar el web y en la opción "Todas las tareas" que aparece al pulsar con el botón derecho seleccionamos "Comprobar extensiones Web":

Veremos como comprueba y repara si es el caso los errores que se hubiesen producido. En ocasiones esta reparación no es suficiente y hay "Quitar las extensiones" y volver a instalarlas con estas opciones para rehabilitarlo. Pero en general es un módulo que suele presentar dificultades.

Nota.-Todas estas configuraciones son a nivel de sitios Web, no de servidor Web.

4.2.3. Optimizar el rendimiento del servidor Web

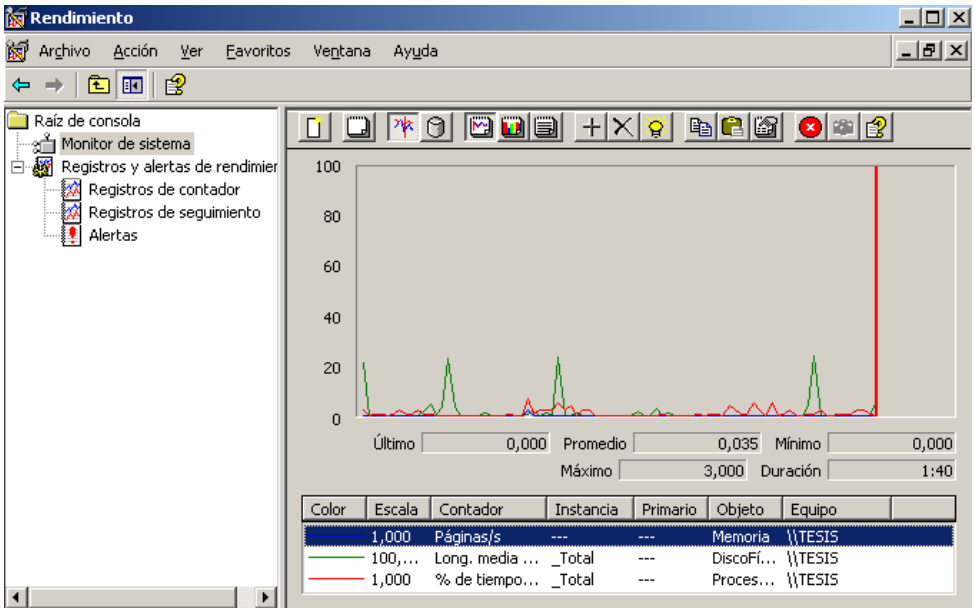

Figura 61Muestra del trafico del servidor mediante el monitor del sistema

Deshabilitar servicios no esenciales

26. Deshabilite los servicios de Windows Server 2003 que no sean necesarios para un servidor Web dedicado. Para ello, siga estos pasos:

1.Haga clic en Inicio, seleccione Programas**,** Herramientas administrativas y, a continuación, haga clic en Administración de equipos.

2. En Administración de equipos (local)**,** expanda Servicios y Aplicaciones y, a continuación, haga clic en Servicios.

En la columna Estado, cada servicio que está en ejecución tiene la etiqueta "Iniciado". En un servidor Web dedicado no se requieren los servicios siguientes:

- Servicio de alerta
- Portafolios
- Examinador de equipos
- Cliente DHCP
- Servidor DHCP
- Servicio de fax

- Replicación de archivos
- Monitor de infrarrojos
- Conexión compartida a Internet
- Messenger
- Escritorio remoto compartido de NetMeeting
- DDE de red
- DSDM de DDE de red
- NetBIOS de NWLink
- NWLink IPX/SPX
- Cola de impresión
- Servicio de ayuda TCP/IP NetBIOS
- Telefonía
- Telnet
- Sistema de alimentación ininterrumpida

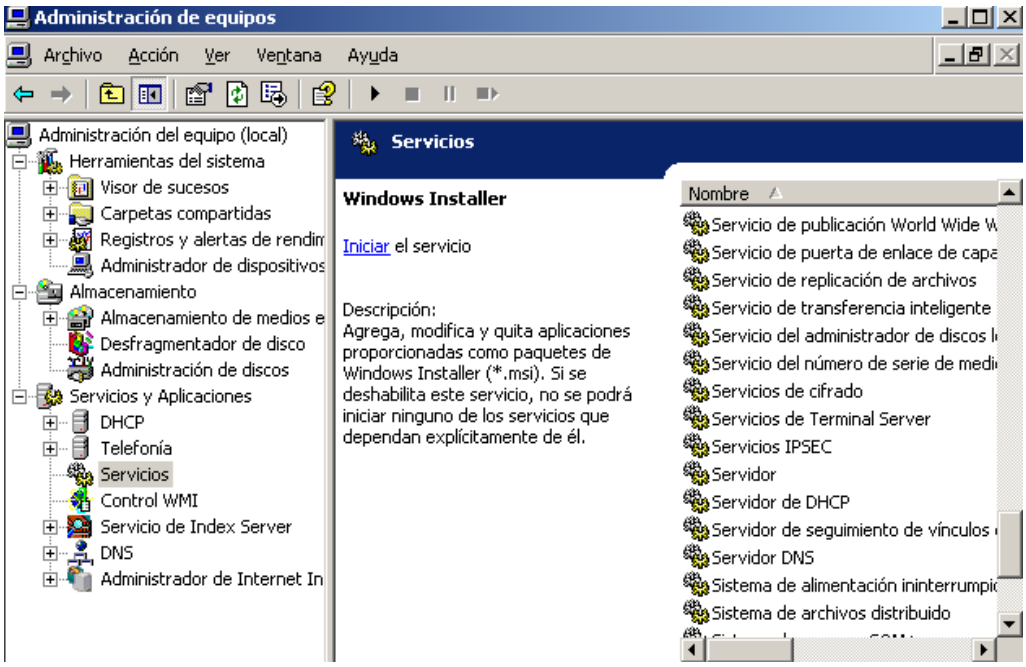

Figura 62 Servicios de Windows

3. Tome nota de los servicios que dependen de cada servicio que desea deshabilitar. Para ello, siga estos pasos:

a. Haga doble clic en el servicio que desea deshabilitar.

b. Haga clic en la ficha Dependencias.

c. En la lista Este servicio depende de los siguientes componentes del sistema, tome nota de los servicios de los que depende este servicio.

d. En la lista Los siguientes componentes del sistema dependen de este servicio, tome nota de los servicios que no se pueden iniciar sin este servicio.

e. Haga clic en Aceptar.

4. Deshabilite los servicios que ha seleccionado, uno por uno. Para ello, siga estos pasos:

a. Haga clic con el botón secundario del *mouse* (ratón) en el servicio que desea deshabilitar y, a continuación, haga clic en Propiedades.

b. En la lista Tipo de inicio, haga clic en Deshabilitado.

c. Si desea detener el servicio inmediatamente, haga clic en Detener. Si se abre el cuadro de diálogo Detener otros servicios, tome nota de los servicios dependientes que también se detendrán y haga clic en Sí.

d. Haga clic en Aceptar.

5.Repita el paso 4 para deshabilitar los demás servicios no esenciales.

Notas

Compruebe que el servidor Web funciona correctamente después de deshabilitar cada uno de los servicios para asegurarse de no deshabilitar un servicio que desea continuar utilizando.

- Si el servidor de Servicios de Internet Information Server (IIS) forma parte de un dominio de Windows Server 2003, debe tener en el sistema el servicio de ayuda TCP/IP para aplicar al equipo la Directiva de grupos correctamente.
- Al deshabilitar el cliente DHCP, dicho cliente detiene el registro dinámico DNS. Esto deshabilita el protocolo de actualización dinámica de DNS y exige que los registros manuales de DNS se agreguen para este cliente en el servidor DNS.

Optimizar el rendimiento de datos para aplicaciones de red

27. Ejecute el código paginable de proceso de IIS 6.0 en la memoria de trabajo. Para ello, siga estos pasos:

1.En el Explorador de Windows, haga clic con el botón secundario del *mouse* en Mis sitios de red y, a continuación, haga clic en Propiedades.

2.Haga clic con el botón secundario del *mouse* en la conexión de área local que desea optimizar y, a continuación, haga clic en Propiedades.

3.En la lista Esta conexión utiliza los siguientes elementos, haga clic (sin desactivar su casilla de verificación) en Compartir impresoras y archivos para redes Microsoft y, a continuación, haga clic en Propiedades.

4.Haga clic en Maximizar el rendimiento para aplicaciones de red, en Aceptar y, finalmente, en Cerrar.

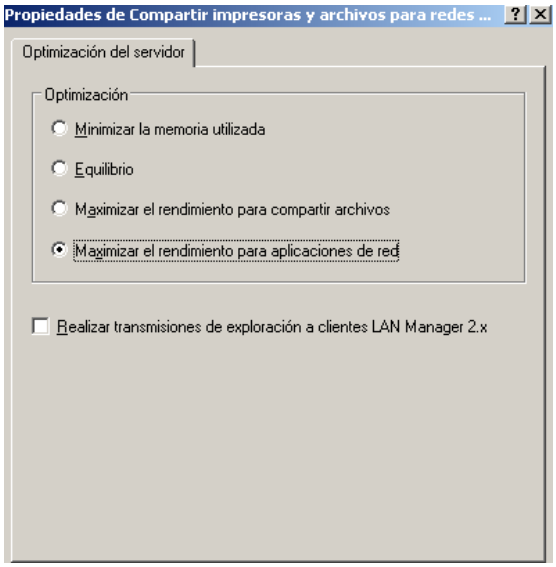

Figura 63 Pantalla para maximizar rendimiento en aplicaciones de red

Optimizar el rendimiento para los servicios en segundo plano

28. El proceso de IIS 6.0 (Inetinfo.exe) se ejecuta como servicio en segundo plano. Para aumentar el rendimiento de los servicios en segundo plano, siga estos pasos:

1.Haga clic en Inicio**,** Panel de control y, después, en Sistema.

2.Haga clic en la ficha Avanzadas y, después, haga clic en Configuración bajo Rendimiento.

3.Haga clic en la ficha Avanzadas, en Servicios en segundo plano y, a continuación, en Aceptar dos veces.

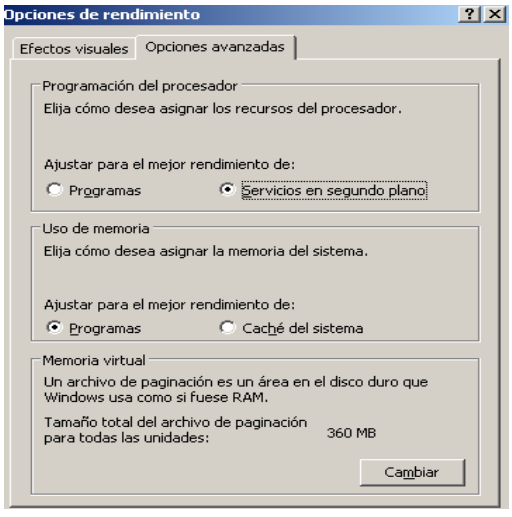

Figura 64 Pantalla para optimizar servicios en segundo plano

Habilitar el límite de ancho de banda

29. Limite el ancho de banda de la red disponible para los distintos sitios Web. Para ello, siga estos pasos:

1.Haga clic en Inicio, seleccione Programas**,** Herramientas administrativas y, a continuación, haga clic en Administrador de servicios Internet**.**
2.Expanda *nombre del servidor*, donde *nombre del servidor* es el nombre del servidor Web.
3.Haga clic con el botón secundario del *mouse* en el sitio Web que desea modificar y, a continuación, haga clic en Propiedades**.**
4.Haga clic en la ficha Rendimiento y, a continuación, active la casilla de verificación Limitar el ancho de banda de red total disponible para este sitio Web.
5.En el cuadro Ancho de banda máximo escriba el nuevo valor y, a continuación, haga clic en Aceptar**.**
6.Cierre el Administrador de servicios Internet.

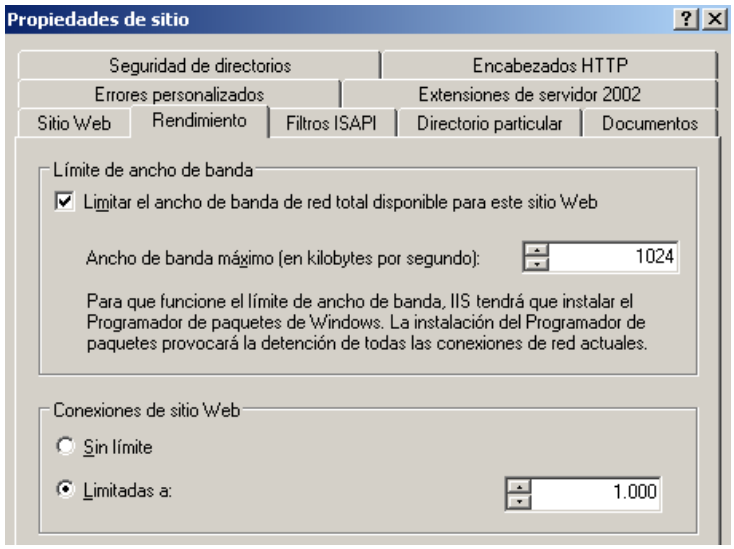

Figura 65 Habilitar limte de ancho de banda

Habilitar la supervisión de la CPU

30. Habilite la supervisión de la CPU para supervisar y apagar automáticamente los procesos de trabajo que consumen grandes cantidades de tiempo de CPU. Para habilitar la supervisión de la CPU, siga estos pasos:

1.En el Administrador de Internet Information Services (IIS), expanda el equipo local, expanda la carpeta Grupos de aplicaciones, haga clic con el botón secundario del *mouse* en el grupo de aplicaciones para el que desea habilitar las cuentas de la CPU y, a continuación, haga clic en Propiedades**.**
2.Haga clic en la ficha Rendimiento y active la casilla de verificación Habilitar la supervisión de la CPU**.**

3.En el cuadro Uso máximo de CPU haga clic en las flechas arriba y abajo para establecer el porcentaje máximo de CPU que desea que utilice el grupo de aplicaciones.

Si el grupo de aplicaciones utiliza más que el máximo designado, IIS genera un mensaje de error en el registro Sucesos de Windows.

4.En el cuadro Actualizar los indicadores de uso de CPU (en minutos) haga clic en las flechas arriba y abajo para establecer el índice de actualización.

5. En el cuadro Acción realizada cuando el uso de CPU supera el máximo haga clic en la acción adecuada para el grupo de aplicaciones designado:

- Haga clic en No se requiere acción para que IIS genere un error en el Registro de sucesos de Windows cuando el grupo de aplicaciones designado llegue al uso de CPU máximo.

- Haga clic en Apagar para cerrar el grupo de aplicaciones. Haga clic en Apagar para detener la aplicación con problemas finalizando su proceso de trabajo de host.

6.Haga clic en Aplicar y, después, en Aceptar**.**

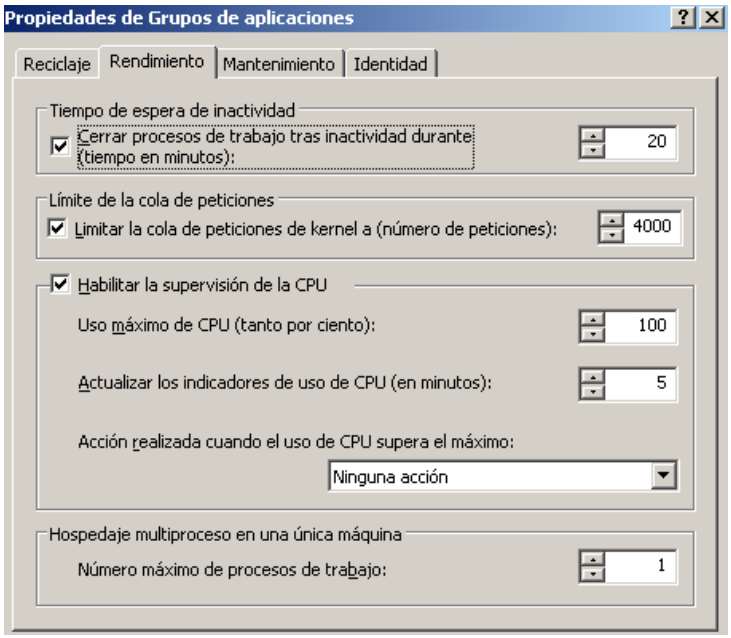

Figura 66 Habilitar supervision de la CPU

Limitar las conexiones al sitio Web

31. Limite el número de conexiones disponibles para cada uno de los sitios Web. Para ello, siga estos pasos:

1.Inicie el Administrador de servicios Internet.

2.Expanda nombre del servidor, donde *nombre del servidor* es el nombre del servidor Web.

3.Haga clic con el botón secundario del *mouse* en el sitio Web que desea limitar y, a continuación, haga clic en Propiedades**.**

4.Haga clic en la ficha Rendimiento y, después, haga clic en Limitadas a**.**

5.En el cuadro Limitadas a**,** escriba el número de conexiones que desea permitir.

Nota

Cada cliente conectado utiliza unas cuatro conexiones simultáneas. Por ejemplo, un límite de conexiones de 200 permite que unos 50 usuarios tengan acceso al sitio Web.

6.Haga clic en Aceptar y cierre el Administrador de servicios de Internet.

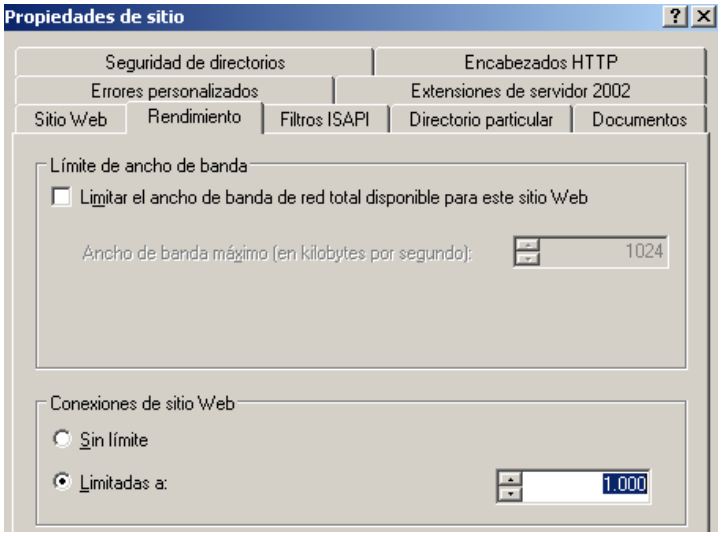

Figura 67 Límite de Conexiones al sitio Web

Utilizar Mantenimiento de conexiones HTTP abiertas

32. De manera predeterminada, el uso de Mantenimiento de conexiones HTTP abiertas está habilitado. Para comprobar que Mantenimiento de conexiones HTTP abiertas está habilitado, siga estos pasos:

1.Inicie el Administrador de servicios Internet.

2.Expanda *nombre del servidor*, donde *nombre del servidor* es el nombre del servidor Web.

3.Haga clic con el botón secundario del *mouse* en el sitio Web que desea modificar y, a continuación, haga clic en Propiedades**.**

4.Haga clic en la ficha Sitio Web, active la casilla de verificación Habilitar mantenimiento de conexiones HTTP abiertas y, a continuación, haga clic en Aceptar.

5.Cierre el Administrador de servicios Internet.

Configuración de la compresión HTTP

Agregar la extensión de servicio web

33. Como primer punto nos dirigiremos a la consola de administración del IIS 6, dentro del Servidor de Internet hacemos clic en la carpeta Extensiones de Servicio Web, aquí tendremos que agregar una extensión ISAPI para la compresión.

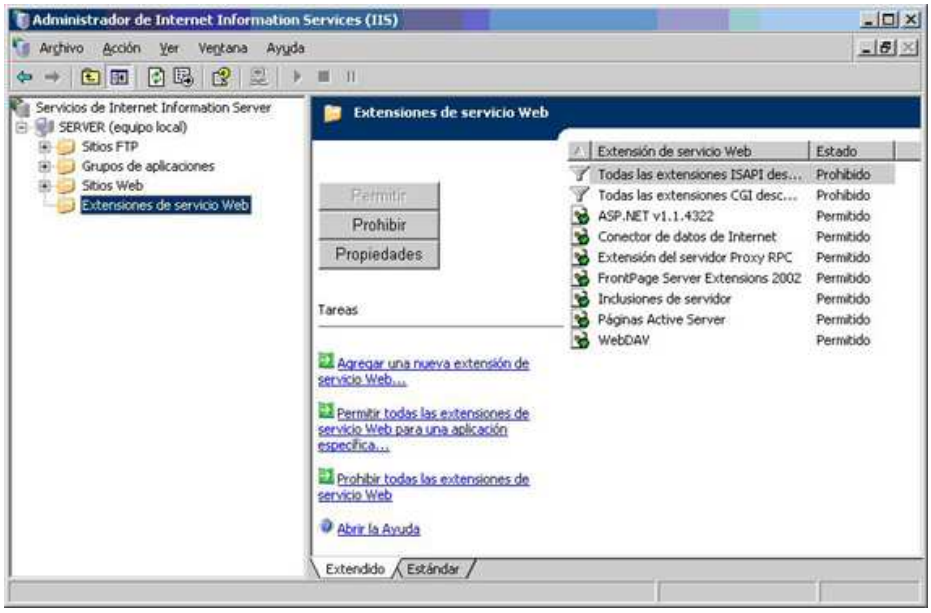

Figura 68 Extensión de servicio Web para la compresión

Seleccionamos **"**Agregar una nueva extensión de servicio Web…**",** allí se desplegará la siguiente pantalla, en la cual tendremos que colocar el nombre que deseemos, en este caso es **"**Compresión**",** después deberemos agregar los archivos necesarios, para ello presionamos **"**Agregar**",** y buscamos el archivo gzip.dll, que se encuentra generalmente en "%windir%\SYSTEM32\inetsrv\gzip.dll**",** por último establecemos el estado de la extensión a Permitido y aceptamos los cambios. Luego de ello deberíamos tener establecido el filtro ISAPI correspondiente para realizar la compresión de los archivos de nuestro sitio.

Figura 69 Agregar extensión de servicio web

Habilitar la compresión estática y dinámica

34. El paso siguiente es habilitar la compresión, para hacerlo buscamos la carpeta Sitios Web y hacemos clic con el botón derecho sobre él, en el menú contextual seleccionamos Propiedades, inmediatamente se desplegará la pantalla de Propiedades, allí elegimos la ficha Servicios. Aquí tendremos que seleccionar las casillas "Comprimir archivos de aplicación" y "Comprimir archivos estáticos", con este último también se mostrará la ubicación de los archivos comprimidos, la cual podremos cambiar de ser necesario; asimismo, se habilita la opción de limitar el tamaño que pueden ocupar los archivos estáticos comprimidos.

Hasta aquí hemos realizado las operaciones posibles sobre la consola de administración, para realizar algunas configuraciones mas detalladas es necesario recurrir a la metabase.

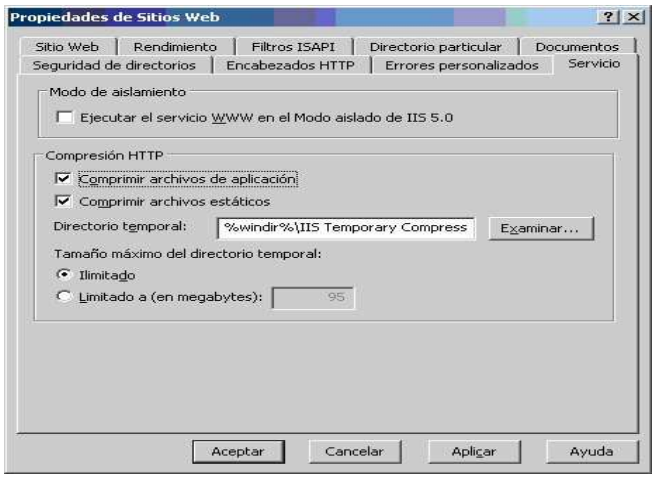

Figura 70 Asignar Compresión a los sitios Web

Hacer copia de seguridad

35. Antes de hacer cualquier cambio en la metabase, debemos hacer un *backup* de la misma. Para realizarlo, deberemos dirigirnos a la consola de administración del IIS, haciendo clic con botón derecho en máquina local (la que corresponda a localhost), luego seleccionar "Todas las tareas", y después hacer clic en "Realizar o restaurar copia de seguridad de configuración …"; allí aparecerá la pantalla de Copia de seguridad / Restauración de la configuración.

Seleccionemos "Crear copia de seguridad", y elijamos un nombre para la copia. También podremos encriptar la misma, pero deberemos en este caso asignarle una contraseña. Una vez hecho el *backup*, debemos tener en cuenta otra cuestión: sería conveniente detener los servicios del IIS mientras se hacen los cambios, de esta manera se evitará problemas en la metabase mientras realizamos los cambios.

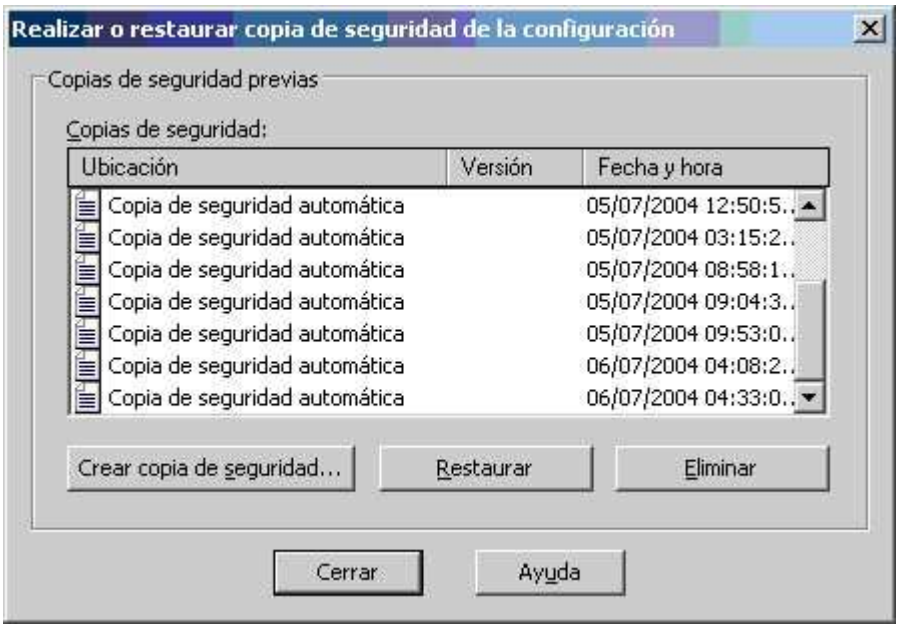

Figura 71 Crear copia de la metabase

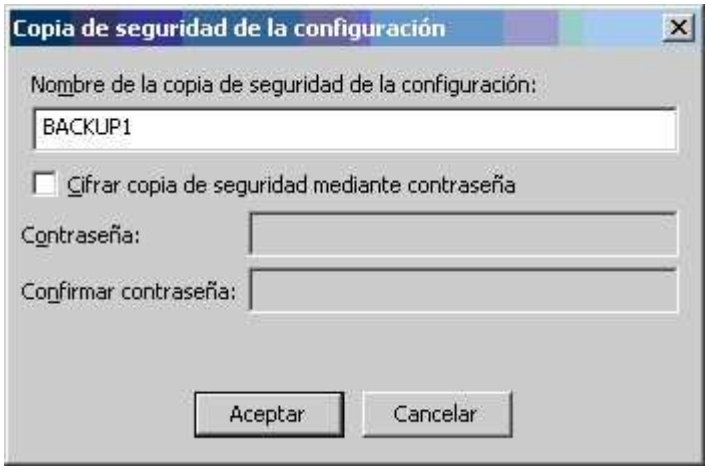

Figura 72 Asignar nombre a la Copia de la Metabase

Nota.-La metabase se encuentra en la ruta
WINDOWS/System32/inetsrv/Metabase.xml.
Para editar la metabase se deben detener cada uno de los servicios del Internet Information Server, para que los cambios que se realicen en la metabase puedan guardarse.

Editar la metabase

36. Para realizar la configuración dentro de la metabase tenemos que saber a dónde nos dirigimos, la entrada que nos interesa es:
<IIsCompressionScheme Location="/LM/W3SVC/Filters/Compression/gzip"
Abre el archivo Metabase.xml con algún editor de texto, luego busca la entrada anterior; veremos que está compuesta por varias propiedades, describiremos las más importantes:

- HcDoDynamicCompression="TRUE", cuando está configurada en *True* significa que está habilitada la compresión de archivos dinámicos. Una propiedad similar es: DoDynamicCompression, sirve para aplicar compresión dinámica a directorios y archivos particulares.
- HcDoOnDemandCompression="TRUE", cuando está configurada en *True* significa que si está habilitada la compresión de archivos estáticos, se enviará al cliente una versión comprimida del archivo, independientemente de si el archivo se encontró o no en el directorio de archivos comprimidos.
- HcDynamicCompressionLevel="9", establece el nivel de compresión de los archivos dinámicos; números altos significan mayores niveles de compresión, pero también implican mayor uso de CPU y de memoria (0-10).
- HcFileExtensions="htm
  html
  txt"
  Aquí debemos colocar las extensiones de archivos estáticos, por ejemplo: *htm*, *html*, *txt*, *doc*, *pdf*, etc. que serán comprimidas si está habilitada la compresión estática.
- HcOnDemandCompLevel="9", establece el nivel de compresión de los archivos estáticos cuando la compresión en demanda está activada, números altos significan mayores niveles de compresión, pero también implican mayor uso de CPU y de memoria (0-10).
- HcScriptFileExtensions="asp
  aspx

dll

js

exe"

Aquí debemos colocar las extensiones de archivos dinámicos que serán comprimidos, en forma predeterminada están: *asp*, *dll*, *exe*, si necesitamos soporte para ASP .NET deberemos agregar *aspx*, por supuesto funciona si está habilitada la compresión dinámica.

El resultado final debe lucir como sigue:

< IIsCompressionScheme Location ="/LM/W3SVC/Filters/Compression/gzip"
HcCompressionDll=”%windir%\system32\inetsvr\gzip.dll”HcCreateFlags=”1”
HcDoDynamicCompression=”TRUE”
HcDoOnDemandCompression=”TRUE”
HcDoStaticCompression=”TRUE”HcDynamicCompressionLevel=”9”
HcFileExtensions=”htm html txt”
HcOnDemandCompLevel=”9”
HcPriority=”1”
HcScriptFileExtensions=”asp aspx dll js exe”

Además, existe otra entrada en la metabase que es interesante conocer y es la siguiente:

< IIsCompressionSchemes Location ="/LM/W3SVC/Filters/Compression/Parameters"
HcCacheControlHeader=”max-age=86400”
HcCompressionBufferSize=”8192”
HcCompressionDirectory=”%windir%\IIS Temporary Compressed Files”
HcDoDiskSpaceLimiting=”FALSE”
HcDoDynamicCompression=”TRUE”
HcDoOnDemandCompression=”TRUE”
HcDoStaticCompression=”TRUE”
HcExpiresHeader=”Wed,01 Jan 1997 12:00:00 GMT”
HcFilesDeletedPerDiskFree=”256”
HcloBufferSize=”8192”
HcMaxDiskSpaceUsage=”99614720”
HcMaxQueueLength=”1000”
HcMinFileSizeForComp=”1”
HcNoCompressionForHttp10=”TRUE”

HcNoCompressionForProxies=”TRUE”
HcNoCompressionForRange=”FALSE”
HcSendCacheHeaders=”FALSE”
Algunas de las propiedades que aparecen aquí también pueden ser modificadas desde la consola de administración:

- HcCompressionDirectory="%windir%\IIS Temporary Compressed Files", indica el directorio en donde serán guardados los archivos temporales comprimidos.
- HcDoDiskSpaceLimiting="FALSE", cuando está en *False* no limita el espacio de los archivos temporales comprimidos.
- HcExpiresHeader="Wed, 01 Jan 1997 12:00:00 GMT", se coloca una fecha obsoleta para que los *proxys* no guarden en caché el archivo.
- HcMaxDiskSpaceUsage="99614720", tamaño máximo en bytes que pueden ocupar las copias comprimidas de los archivos estáticos.

Una vez hechas las modificaciones en el archivo Metabase.xml, guardaremos los cambios, y procederemos a iniciar los servicios del IIS 6. En este punto la compresión debería estar funcionando correctamente.

Sugerencia:
Para comprobar que la compresión efectivamente funcione, existen sitios que tienen una herramienta en línea, en la cual generalmente hay que proporcionar la URL del sitio a chequear para luego generar un reporte, el cual nos informa si existe compresión o no, y de cuánto es la tasa de compresión, además de otros datos. Algunos de estos sitios son:
http://www.pipeboost.com/
http://www.xcompress.com/
http://www.port80software.com/
http://www.turboiis.com/

También podemos realizar una prueba utilizando el monitor de rendimiento de Windows; deberíamos incluir las siguientes variables: (a) uso del procesador, (b) uso de la memoria y (c) cantidad de bytes enviados por la interfaz de red.
A continuación se muestran dos gráficos con dos variables: *procesador*, que está indicada con rojo, y bytes enviados por la *interfaz de red*, indicada con azul.

Compresión deshabilitada

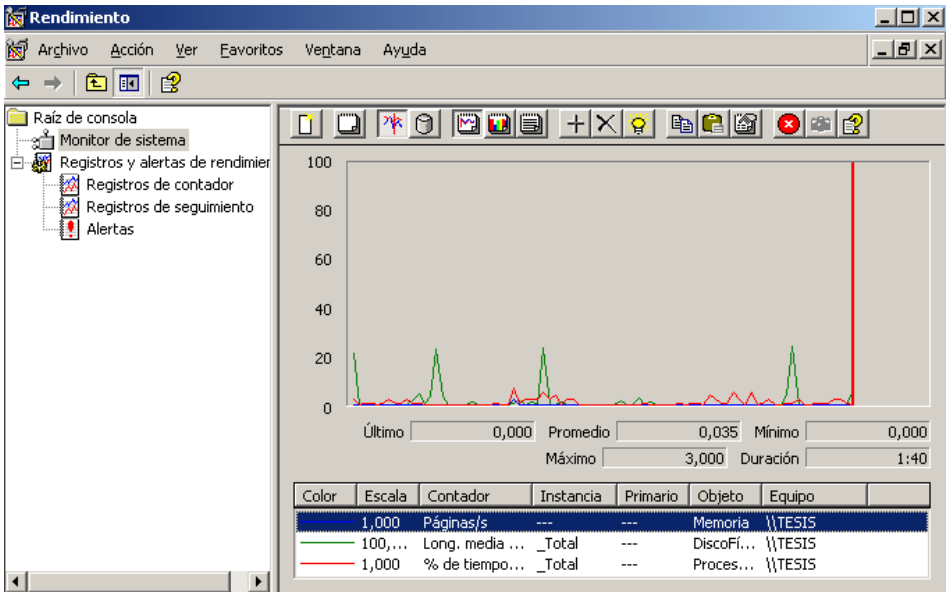

Figura 73 Compresión Deshabilitada

Compresión habilitada

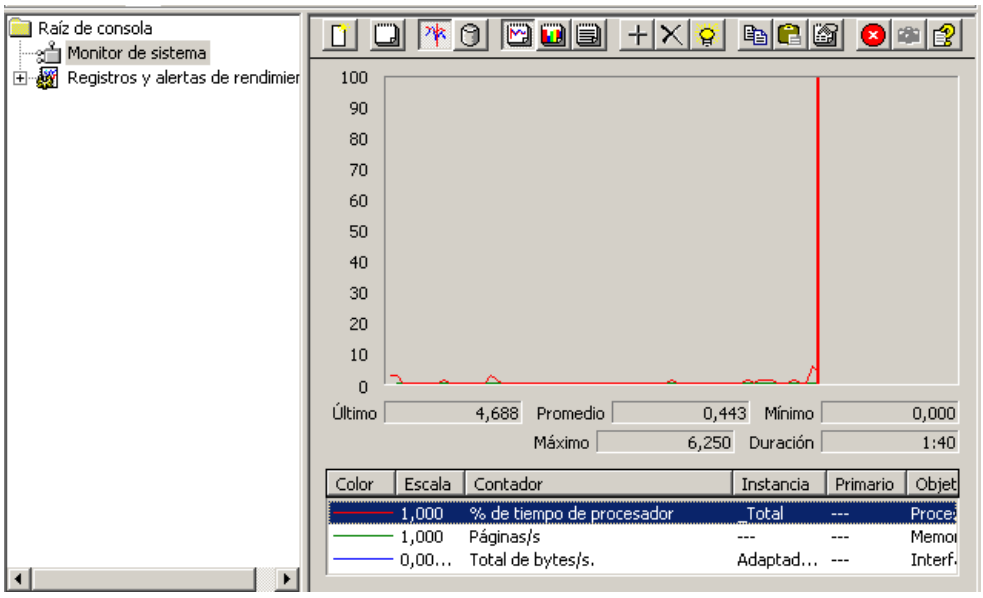

Figura 74 Compresión Habilitada

Uno de los gráficos corresponde a las mediciones realizadas cuando el sitio estaba sin compresión, y el otro cuando la compresión estaba activada. Se puede apreciar, comparándolos, la diferencia existente en los bytes enviados en las diferentes respuestas.

Hay que tener en cuenta al realizar las mediciones si estamos trabajando con ASP .NET: esto es porque, después de cada reinicio de los servicios del IIS 6, se vuelven a compilar nuevamente las páginas al ser llamadas por primera vez. Por ello, el monitor de rendimiento nos podría indicar un gran uso de procesador y memoria, lo que no

tiene nada que ver con la compresión en este caso, sino con la sobrecarga de dicha compilación.

Desventaja de la Utilización de la compresión http.

Se aconseja utilizar compresión HTTP cuando el ancho de banda existente es escaso, ya sea porque el enlace utilizado es lento por naturaleza, no sólo del lado del servidor *web* sino también del lado de los clientes, o porque el sitio *web* tiene un gran tráfico de información. En cualquiera de los casos la compresión es beneficiosa, pero existe una pequeña desventaja: esta técnica insume tiempo de procesamiento.

Sin embargo, esto no debería preocuparnos a menos que la tasa de uso del procesador ya sea muy elevada, en este caso no se recomienda su utilización

4.2.4 Seguridad

Reducción del ámbito de ataque en el servidor Web

Inicie el proceso de aplicación de seguridad en el servidor reduciendo el ámbito de ataque o el campo al que se expone el servidor frente a atacantes potenciales. Por ejemplo, habilite únicamente aquellos componentes, servicios y puertos que sean necesarios para el funcionamiento correcto del servidor.

Selección de componentes y servicios de IIS fundamentales

37. IIS 6.0 incluye subcomponentes y servicios además del servicio WWW como, por ejemplo, los servicios FTP y SMTP. Para reducir el riesgo de ataques cuyo objetivo son servicios y subcomponentes específicos, se recomienda seleccionar únicamente aquellos que los sitios y aplicaciones Web necesiten para funcionar correctamente.

En la siguiente tabla se muestra la configuración recomendada en Agregar o quitar programas para los subcomponentes y servicios de IIS en el servidor Web.

Subcomponente o servicio	**Configuración predeterminada**	**Configuración del servidor Web**
Extensiones de Servicio de transferencia inteligente en segundo plano (BITS)	Deshabilitado	Sin cambio
Archivos comunes	Habilitado	Sin cambio
Servicio FTP	Deshabilitado	Sin cambio
Extensiones de servidor de FrontPage 2002	Deshabilitado	Sin cambio
Administrador de servicios de Internet Information Server	Habilitado	Sin cambio
Impresión de Internet	Deshabilitado	Sin cambio

Servicio NNTP	Deshabilitado	Sin cambio
Servicio SMTP	Habilitado	Deshabilitado
Servicio World Wide Web	Habilitado	Sin cambio

Tabla 13 Configuración recomendada para subcomponentes y servicios IIS

Requisitos

•**Credenciales:** la sesión se debe iniciar como miembro del grupo de administradores en el servidor Web.

•**Herramientas:** Agregar o quitar programas.

Para configurar componentes y servicios de IIS

1. Haga clic en Inicio, Panel de control y, a continuación, en Agregar o quitar programas.
2. Seleccione Agregar o quitar componentes de Windows.
3. En la página Asistente para componentes de Windows, en Componentes, haga clic en Servidor de aplicaciones y, a continuación, en Detalles.
4. Haga clic en Servicios de Internet Information Server (IIS) y, a continuación, en Detalles.
5. Consulte la tabla anterior y seleccione o anule la selección de los componentes y servicios de IIS adecuados. Para ello, active o desactive la casilla de verificación del componente o servicio correspondiente.
6. Siga las instrucciones del Asistente para componentes de Windows para finalizar.

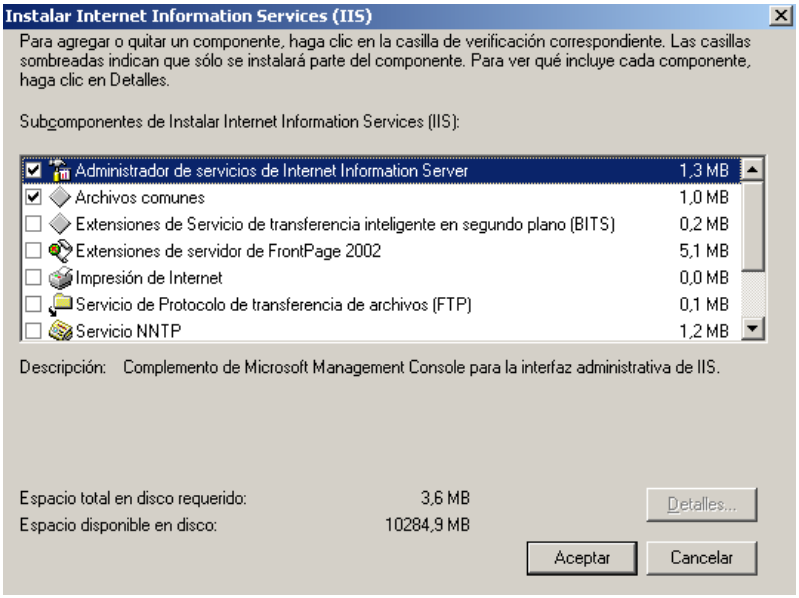

Figura 75 Subcomponentes de IIS

Comprobación de la nueva configuración

Asegúrese de que se ha aplicado la configuración de seguridad adecuada al servidor Web.

Para comprobar que los componentes y servicios de IIS se han seleccionado

•Haga clic en Inicio, Panel de control y, a continuación, en Herramientas administrativas.

El Administrador de Internet Information Server (IIS) aparecerá ahora en el menú de herramientas administrativas.

Habilitación de las extensiones de servicio Web esenciales

38. Un servidor Web que ofrece contenido dinámico necesita extensiones de servicio Web. Cada tipo de contenido se corresponde con una extensión específica. Por razones de seguridad, IIS 6.0 permite habilitar y deshabilitar extensiones de servicio Web independientes, de modo que se habiliten únicamente las necesarias para el contenido.

ADVERTENCIA: no active todas las extensiones de servicio Web. Aunque con ello se asegura la mayor compatibilidad posible con los sitios Web y aplicaciones existentes, el ámbito de ataque del servidor Web aumenta considerablemente. Puede que sea necesario comprobar los sitios y aplicaciones Web por separado, para asegurar que sólo se habilitan las extensiones necesarias.

Supongamos que el servidor Web se configura para proporcionar el archivo Default.asp y su página predeterminada. Aunque se configure la página predeterminada, debe habilitar la extensión de servicio Web Páginas Active Server para poder visualizar la página .asp.

Requisitos

•**Credenciales:** la sesión se debe iniciar como miembro del grupo de administradores en el servidor Web.

•**Herramientas:** Administrador de Internet Information Server (Iis.msc).

Para activar la extensión de servicio Web Páginas Active Server

1. Haga clic en Inicio, Panel de control, Herramientras administrativas y, a continuación, haga doble clic en Administrador de Internet Information Server (IIS).
2. Haga doble clic en el equipo local y, a continuación, en Extensiones de servicio Web.
3. Seleccione Páginas Active Server y, posteriormente, Permitir.

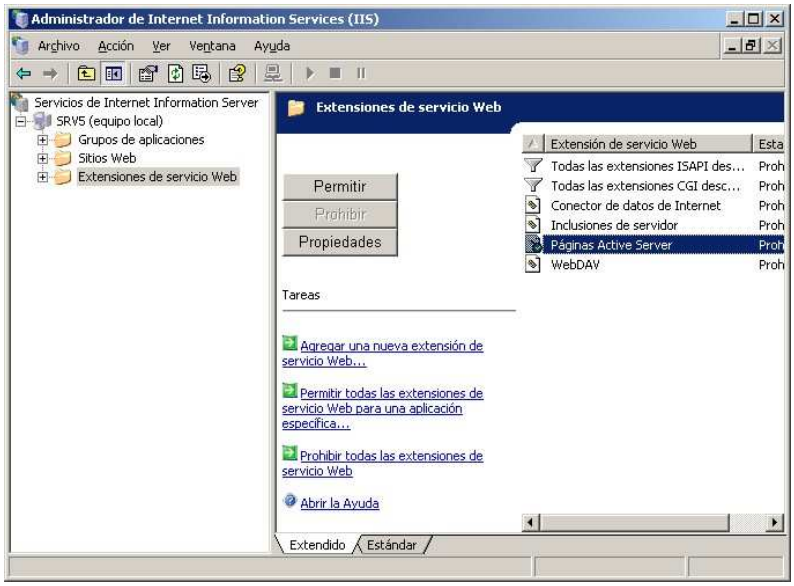

Figura 76 Activacion de Extension para Páginas Active Server

Comprobación de la nueva configuración

Asegúrese de que se ha aplicado la configuración de seguridad adecuada al servidor Web.

Para comprobar que la extensión de servicio Web Páginas Active Server está habilitada

1. Abra un editor de textos, escriba algo y guarde el archivo como Default.asp en el directorio C:\inetpub\wwwroot.
2. En el cuadro Dirección de Internet Explorer, indique la siguiente dirección URL y, a continuación, presione Enter: http://localhost

   El archivo Default.asp aparecerá en el explorador.

Configuración de cuentas

•Deshabilitación de cuentas no utilizadas

•Aislamiento de aplicaciones con grupos de ellas

Deshabilitación de cuentas no utilizadas

Un atacante puede hacer uso de cuentas que no se utilizan y sus privilegios, a fin de obtener acceso al servidor. Es necesario auditar de forma periódica las cuentas locales del servidor y deshabilitar las cuentas que no se empleen. Deshabilite las cuentas en un servidor de prueba antes de hacerlo en un servidor de producción; de este modo, se asegurarán de que no se afecta negativamente el funcionamiento de la aplicación. Si la deshabilitación de la cuenta no causa ningún problema en el servidor de prueba, realice la misma operación en el servidor de producción.

Nota: si decide eliminar una cuenta no utilizada en lugar de deshabilitarla, tenga en cuenta que no se podrá recuperar y que no se pueden eliminar las cuentas Administrador e Invitado. Asimismo, asegúrese de eliminar la cuenta en un servidor de prueba antes de hacerlo en uno de producción.

En esta parte se proporcionan las siguientes instrucciones paso a paso para la eliminación o deshabilitación de cuentas que no se utilizan:

- Deshabilitación de la cuenta de invitado
- Cambio de nombre de la cuenta de administrador
- Cambio de nombre de la cuenta IUSR_*NombreEquipo*

Deshabilitación de la cuenta de invitado

39.-La cuenta de invitado se utiliza cuando se realiza una conexión anónima al servidor Web. Durante una instalación predeterminada de Windows Server 2003, esta cuenta se deshabilita. Para restringir las conexiones anónimas al servidor, compruebe que la cuenta de invitado se ha deshabilitado.

Requisitos

- **Credenciales:** la sesión se debe iniciar como miembro del grupo de administradores en el servidor Web.
- **Herramientas:** Administración de equipos

Para deshabilitar la cuenta de invitado

1. Seleccione Inicio, haga clic con el botón secundario del mouse en Mi PC y, a continuación, haga clic en Administrar.
2. Haga doble clic en Usuarios locales y grupos y, a continuación, haga clic en la carpeta Usuarios. La cuenta de invitado debe aparecer con un icono *X* de color rojo, que indica que está deshabilitada. Si no lo está, continúe con el paso 3 para hacerlo.

Figura 77 Desactivacion de la cuenta de Usuario

3. Haga clic con el botón secundario del mouse en la cuenta de invitado y, a continuación, elija Propiedades.

4. En la ficha General, active la casilla de verificación Cuenta deshabilitada y, a continuación, haga clic en Aceptar.

Ahora la cuenta aparecerá con un icono *X* de color rojo.

Cambio de nombre de la cuenta de administrador

40.-La cuenta de administrador local predeterminada es uno de los objetivos de los usuarios malintencionados debido a sus elevados privilegios en el equipo. Para mejorar la seguridad, se debe cambiar el nombre predeterminado y asignarle una contraseña segura.

Requisitos

•**Credenciales:** la sesión se debe iniciar como miembro del grupo de administradores en el servidor Web.

•**Herramientas:** Mi PC.

Para cambiar el nombre de la cuenta de administrador y asignarle una contraseña segura

1. Seleccione Inicio, haga clic con el botón secundario del mouse en Mi PC y, a continuación, haga clic en Administrar.

2. Haga doble clic en Usuarios locales y grupos y, a continuación, haga clic en la carpeta Usuarios.

3. Haga clic con el botón secundario del mouse en la cuenta de administrador y, a continuación, seleccione Cambiar nombre.

4. Indique un nombre en el cuadro y presione Entrar.

5. En el Escritorio, presione Ctrl+Alt+Supr y haga clic en Cambiar contraseña.

6. Escriba el nuevo nombre para la cuenta de administrador en el cuadro Nombre de usuario.

7. Escriba la contraseña actual en el cuadro Contraseña anterior y una nueva contraseña en el cuadro Contraseña nueva. Indique la nueva contraseña en el cuadro Confirmar contraseña nueva y, a continuación, haga clic en Aceptar.

Advertencia: no utilice el elemento de menú Establecer contraseña en el menú contextual para cambiar la contraseña a no ser que la haya olvidado y no disponga de un disco para restablecer contraseña. Si utiliza este método para cambiar la contraseña de administrador, puede causar la pérdida irreversible de información que haya protegido mediante esta contraseña.

Cambio de nombre de la cuenta IUSR

41.-La cuenta de usuario anónima predeterminada de Internet, IUSR_*NombreEquipo*, se crea durante la instalación de IIS. El valor de *NombreEquipo* será el nombre de NetBIOS del servidor cuando se instale IIS.

Requisitos

- **Credenciales:** la sesión se debe iniciar como miembro del grupo de administradores en el servidor Web.
- **Herramientas:** Mi PC.

Para cambiar el nombre de la cuenta IUSR

1. Seleccione Inicio, haga clic con el botón secundario del mouse en Mi PC y, a continuación, haga clic en Administrar.

2. Haga doble clic en Usuarios locales y grupos y, a continuación, haga clic en la carpeta Usuarios.

3. Haga clic con el botón secundario del mouse en la cuenta IUSR_*NombreEquipo* y, a continuación, seleccione Cambiar nombre.

4. Escriba el nuevo nombre y presione Entrar.

Para cambiar el valor de la cuenta IUSR en la metabase de IIS

1. Haga clic en Inicio, Panel de control, Herramientras administrativas y, a continuación, haga doble clic en Administrador de Internet Information Server (IIS).

2. Haga clic con el botón secundario del mouse en el equipo local y, a continuación, haga clic en Propiedades.

3. Active la casilla de verificación Habilitar la modificación directa de archivos de metabase y, a continuación, haga clic en Aceptar.

4. Explore la ubicación del archivo MetaBase.xml que, de forma predeterminada, es C:\Windows\system32\inetsrv.

5. Haga clic con el botón secundario del mouse en el archivo MetaBase.xml y, a continuación, seleccione Modificar.

6. Busque la propiedad AnonymousUserName y escriba el nuevo nombre de la cuenta IUSR.

7. En el menú Archivo, haga clic en Salir y, a continuación, en Sí.

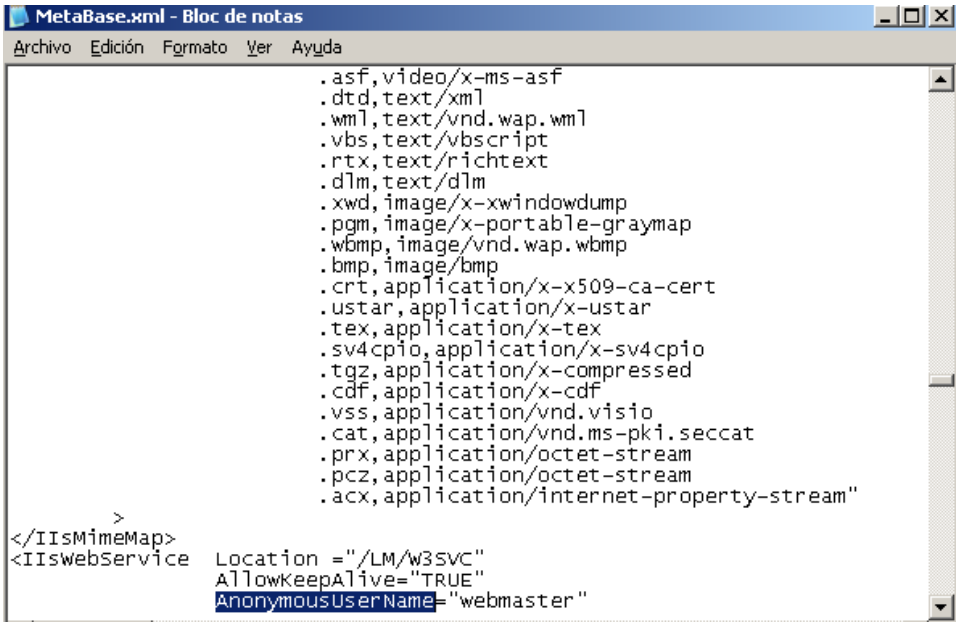

Figura 78 Archivo de la Metabase

Comprobación de la nueva configuración

Asegúrese de que se ha aplicado la configuración de seguridad adecuada al servidor Web.

Para comprobar que una cuenta está deshabilitada

1. Presione Ctrl+Alt+Supr y, a continuación, haga clic en Cerrar sesión para cerrar la sesión del servidor Web.

2. En el cuadro de diálogo Iniciar sesión en Windows, indique el nombre de la cuenta deshabilitada en el cuadro Nombre de usuario, escriba la contraseña de la misma y, a continuación, haga clic en Aceptar.

Aparecerá el siguiente mensaje:

Su cuenta está desactivada. Póngase en contacto con el administrador de su sistema.

Para comprobar el cambio de nombre de una cuenta

1. Presione Ctrl+Alt+Supr y, a continuación, haga clic en Cerrar sesión para cerrar la sesión del servidor Web.
2. En el cuadro de diálogo Iniciar sesión en Windows, indique el nombre de la cuenta con nuevo nombre en el cuadro Nombre de usuario, escriba la contraseña de la misma y, a continuación, haga clic en Aceptar.

   Aparecerá el siguiente mensaje: No se puede iniciar su sesión. Asegúrese de que su nombre de usuario y dominio sean correctos, luego repita su contraseña. Las letras de la contraseña se deben escribir usando mayúsculas y minúsculas correctamente.
3. Haga clic en Aceptar y, a continuación, escriba el nuevo nombre de la cuenta en el cuadro Nombre de usuario.
4. Escriba la contraseña de la cuenta con nuevo nombre y haga clic en Aceptar.

   Se debe poder iniciar la sesión en el equipo con esta cuenta.

Configuración de seguridad para archivos y directorios

Para proteger archivos y directorios importantes se deben emplear controles de acceso seguros. En la mayoría de las situaciones, permitir el acceso a determinadas cuentas resulta más eficaz que denegarlo. El acceso se debe establecer en el nivel de directorio, siempre que sea posible. A medida que los archivos se agregan a la carpeta, heredan permisos de la misma, por lo que no es necesario realizar ninguna acción adicional.
En esta sección se proporcionan las siguientes instrucciones paso a paso para la configuración de seguridad de archivos y directorios:

- Reubicación y definición de permisos para archivos de registro de IIS
- Configuración de permisos de metabase de IIS

Reubicación y definición de permisos para archivos de registro de IIS

42.-Con el fin de aumentar la seguridad de los archivos de registro de IIS, es necesario reubicarlos en una unidad que no sea de sistema y con formato para utilizar el sistema de archivos NTFS. Esta ubicación no debe ser la misma que la del contenido del sitio Web.

Requisitos

- **Credenciales:** la sesión se debe iniciar como miembro del grupo de administradores en el servidor Web.
- **Herramientas:** Mi PC y el Administrador de Internet Information Server (IIS) (Iis.msc).

Para desplazar la ubicación de los archivos de registro de IIS a una partición que no sea de sistema

1. Seleccione Inicio, haga clic con el botón secundario en Mi PC y, a continuación, haga clic en Explorar.
2. Examine la ubicación donde desee reubicar los archivos.
3. Haga clic con el botón secundario del mouse en el directorio de un nivel superior donde desee reubicar los archivos; seleccione Nuevo y, a continuación, Carpeta.
4. Indique un nombre para la carpeta como, por ejemplo, ContosoIISLogs y presione Entrar.
5. Haga clic en Inicio, Panel de control, Herramientras administrativas y, a continuación, haga doble clic en Administrador de Internet Information Server (IIS).
6. Haga clic con el botón secundario del mouse en el sitio Web y, a continuación, haga clic en Propiedades.
7. Haga clic en la ficha Sitio Web y, a continuación, en Propiedades en el marco Habilitar registro.
8. En la ficha Propiedades generales, seleccione Examinar y vaya a la carpeta que se acaba de crear para almacenar los archivos de registro de IIS.
9. Haga clic en Aceptar tres veces.

Nota: si ya dispone de archivos de registro de IIS en la ubicación original en Windows\System32\Logfiles, debe moverlos a la nueva ubicación de forma manual. IIS no realizará la operación por sí mismo.

Para establecer listas de control de acceso (ACL) en archivos de registro de IIS

1. Seleccione Inicio, haga clic con el botón secundario en Mi PC y, a continuación, haga clic en Explorar.
2. Vaya a la carpeta donde se encuentran los archivos de registro.
3. Haga clic con el botón secundario del mouse en la carpeta, seleccione Propiedades y, a continuación, haga clic en la ficha Seguridad.
4. En el panel superior, haga clic en Administradores y compruebe que los permisos del panel inferior se definen como Control total.
5. En el panel superior, haga clic en Sistema y compruebe que los permisos del panel inferior se definen como Control total y, a continuación, haga clic en Aceptar.

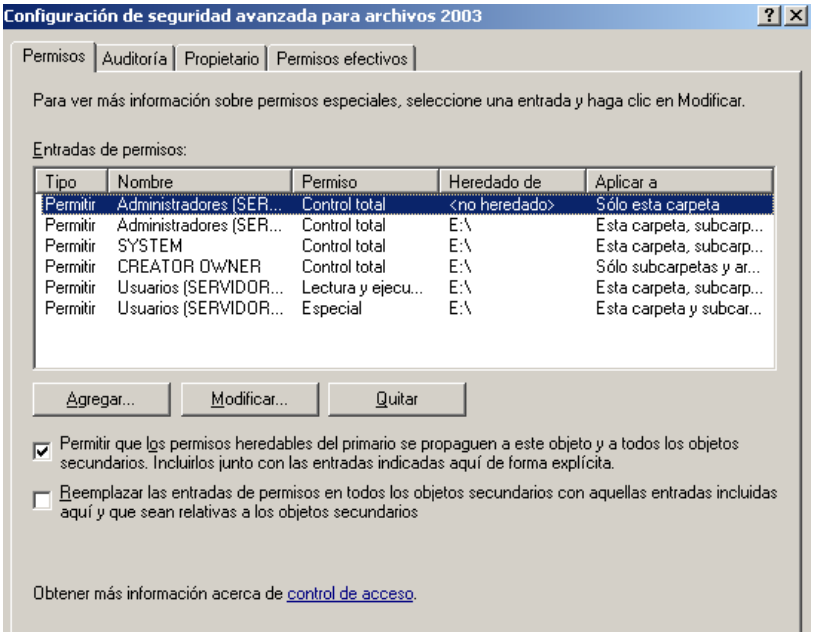

Figura 79 Listas de Control de acceso a los registros de IIS

Comprobación de la nueva configuración

Asegúrese de que se ha aplicado la configuración de seguridad adecuada al servidor Web.

Para comprobar que los archivos de registro se desplazan y los permisos se definen

1. Haga clic en Inicio, Buscar y seleccione Archivos o carpetas.
2. Indique el nombre completo o parcial del archivo en el cuadro Buscar archivos con el nombre como, por ejemplo, LogFiles. Seleccione una ubicación en el cuadro Buscar en y, a continuación, haga clic en Buscar ahora. La búsqueda devuelve la nueva ubicación de los archivos de registro.
3. Presione Ctrl+Alt+Supr y, a continuación, haga clic en Cerrar sesión.
4. Inicie la sesión en el servidor Web utilizado una cuenta que no tenga permiso de acceso a los archivos de registro.
5. Seleccione Inicio, haga clic con el botón secundario del mouse en Mi PC, haga clic en Explorar y, a continuación, examine el directorio LogFiles.
6. Haga clic con el botón secundario en el directorio LogFiles y, posteriormente, seleccione Abrir.

Aparecerá el siguiente mensaje:

•Acceso denegado.

Configuración de permisos de metabase de IIS

43.- La metabase de IIS es un archivo XML que incluye la mayor parte de la información de configuración de IIS.

Requisitos

•**Credenciales:** la sesión se debe iniciar como miembro del grupo de administradores en el servidor Web.

•**Herramientas:** Mi PC y el archivo MetaBase.xml.

Para restringir el acceso al archivo MetaBase.xml

1. Seleccione Inicio, haga clic con el botón secundario en Mi PC y, a continuación, haga clic en Explorar.
2. Examine el archivo Windows\System32\Inetsrv\MetaBase.xml, haga clic con el botón secundario del mouse en el mismo y, a continuación, seleccione Propiedades.
3. Haga clic en la ficha Seguridad, confirme que sólo los miembros del grupo de administradores y la cuenta LocalSystem tienen acceso de control total a la metabase, elimine todos los demás permisos de archivo y haga clic en Aceptar.

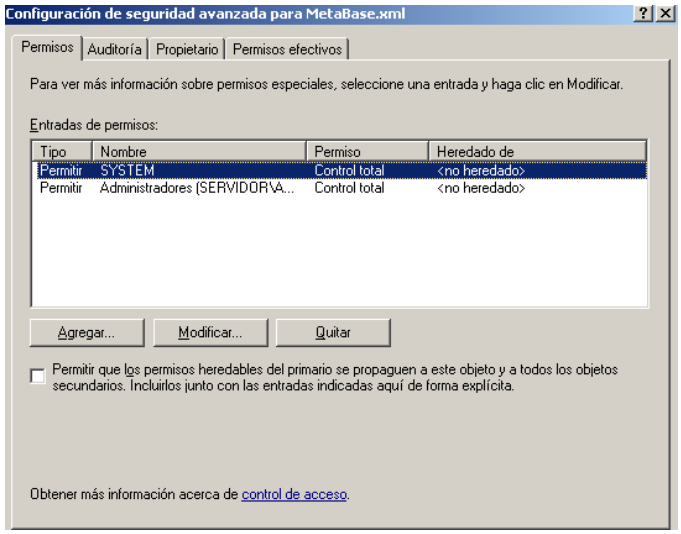

Figura 80 Configuración de seguridad avanzada para Metabase

Comprobación de la nueva configuración

Asegúrese de que se ha aplicado la configuración de seguridad adecuada al servidor Web.

Para comprobar el acceso restringido al archivo MetaBase.xml

1. Presione Ctrl+Alt+Supr y, a continuación, haga clic en Cerrar sesión.
2. Inicie la sesión en el servidor Web utilizado una cuenta que no tenga permiso de acceso al archivo MetaBase.xml.
3. Seleccione Inicio, haga clic con el botón secundario en Mi PC, haga clic en Explorar y, continuación, vaya a la ubicación de MetaBase.xml.
4. Haga clic con el botón secundario del mouse en el archivo MetaBase.xml y, posteriormente, seleccione Abrir.

Aparecerá el siguiente mensaje:

Acceso denegado.

Seguridad en sitios Web y directorios virtuales

Los directorios raíz Web y los directorios virtuales se deben reubicar en una partición que no sea del sistema para ayudar a protegerse de los ataques transversales de directorio. Estos ataques permiten al atacante ejecutar herramientas y programas del sistema operativo. Debido a que no es posible atravesar unidades, la reubicación del contenido del sitio Web en otra unidad ofrece protección adicional frente a este tipo de ataques.

En esta sección se proporcionan las siguientes instrucciones paso a paso para dotar de seguridad a los sitios Web y directorios virtuales:

•Desplazamiento del contenido del sitio Web a una unidad que no es del sistema

•Configuración de permisos del sitio Web

Desplazamiento del contenido del sitio Web a una unidad que no es del sistema

44.- No utilice el directorio predeterminado \Inetpub\Wwwroot como ubicación para el contenido del sitio Web. Por ejemplo, si el sistema se instala en la unidad C: puede mover el directorio del contenido y sitio a la unidad D: para poder eliminar los riesgos asociados con los ataques transversales de directorio, en los que un atacante intenta examinar la estructura de directorio de un servidor Web. Asegúrese de comprobar que todos los directorios virtuales señalan a la nueva unidad.

Requisitos

•**Credenciales:** la sesión se debe iniciar como miembro del grupo de administradores en el servidor Web.

•**Herramientas:** Administrador de Internet Information Server (Iis.msc) y un símbolo del sistema.

Para desplazar el contenido del sitio Web a una unidad que no es del sistema

1. Haga clic en Inicio, Panel de control, Herramientras administrativas y, a continuación, haga doble clic en Administrador de Internet Information Server (IIS).
2. Haga clic con el botón secundario del mouse en el sitio Web que incluye el contenido que desee mover y, a continuación, seleccione Detener.
3. Haga clic en Inicio, Ejecutar, escriba cmd en el cuadro Abrir y, a continuación, seleccione Aceptar.
4. Escriba el siguiente comando en el símbolo del sistema:

   xcopy c:\inetpub\wwwroot*Sitio*

   Unidad:\wwwroot*Sitio*/s/i/o

   En el comando anterior, sustituya

•*Sitio* >con el nombre de su sitio Web.

- *Unidad* con el nombre de la nueva unidad; por ejemplo, D.

5. Vuelva al complemento Administrador de Internet Information Server (IIS).

6. Haga clic con el botón secundario del mouse en el sitio Web y, a continuación, seleccione Propiedades.

7. Haga clic en la ficha Directorio principal, Directorio virtualo Directorio, según el tipo de aplicación seleccionada y, a continuación, indique la nueva ubicación del directorio en el cuadro Ruta local y, posteriormente, seleccione Aceptar.

O bien

Vaya a la nueva ubicación del directorio en el que acaba de copiar los archivos y seleccione Aceptar.

8. Haga clic con el botón secundario del mouse en el sitio Web y, a continuación, seleccione Inicio.

Comprobación de la nueva configuración

Asegúrese de que se ha aplicado la configuración de seguridad adecuada al servidor Web.

Para comprobar que el contenido del sitio Web se ha movido a una unidad que no es del sistema

1. Haga clic en Inicio, Buscar y seleccione Archivos o carpetas.
2. Indique el nombre completo o parcial del archivo en el cuadro Buscar archivos con el nombre. Seleccione una ubicación en el cuadro Buscar en y, a continuación, haga clic en Buscar ahora.

La búsqueda devuelve la lista de archivos desplazados a su nueva ubicación, así como la ubicación original.

Para eliminar el contenido del sitio Web de la unidad del sistema

- Vaya al directorio C:\Inetpub\Wwwroot*Sitio* y, a continuación, elimine los archivos desplazados a una unidad que no sea del sistema.

Comprobación de la nueva configuración

Asegúrese de que se ha aplicado la configuración de seguridad adecuada al servidor Web.

Para comprobar que el contenido del sitio Web se ha eliminado de la unidad del sistema

1. Haga clic en Inicio, Buscar y seleccione Archivos o carpetas.
2. Indique el nombre completo o parcial del archivo en el cuadro Buscar archivos con el nombre. Seleccione una ubicación en el cuadro Buscar en y, a continuación, haga clic en Buscar ahora.

La búsqueda sólo devuelve la lista de archivos desplazados a su nueva ubicación.

Configuración de permisos del sitio Web

Los permisos de acceso se pueden configurar para el servidor Web para directorios, archivos y sitios específicos. Estos permisos se aplican a todos los usuarios independientemente de sus derechos de acceso concretos.

Configuración de permisos en directorios del sistema de archivos

45. IIS 6.0 depende de los permisos NTFS para ayudar a proteger los archivos y directorios independientes de accesos no autorizados. A diferencia de los permisos del sitio Web, aplicados a los usuarios que intentan tener acceso al sitio, se pueden utilizar los permisos NTFS para definir qué usuarios pueden tener acceso al contenido y el modo en que se permite su manipulación. Para mejorar la seguridad, utilice los permisos de sitio Web y NTFS.

Las listas de control de acceso (ACL) indican qué usuarios o grupos tienen permisos para tener acceso o modificar un archivo concreto. En lugar de establecer listas ACL en cada archivo, cree nuevos directorios para cada tipo de archivo, establezca listas ACL en cada directorio y, posteriormente, permita que los archivos hereden permisos del directorio en el que residen.

Requisitos

- **Credenciales:** la sesión se debe iniciar como miembro del grupo de administradores en el servidor Web.
- **Herramientas:** Mi PC y el Administrador de Internet Information Server (IIS) (Iis.msc).

Para desplazar el contenido del sitio Web a una carpeta independiente

1. Seleccione Inicio, haga clic con el botón secundario en Mi PC y, a continuación, haga clic en Explorar.
2. Examine la carpeta que incluya el contenido y, a continuación, haga clic en la carpeta de nivel superior del contenido del sitio Web.
3. En el menú Archivo, haga clic en Nuevo y, a continuación, en Carpeta para crear una nueva carpeta en el directorio de contenido del sitio Web.
4. Indique el nombre de la carpeta y presione Entrar.
5. Presione Ctrl y, a continuación, seleccione cada una de las páginas que desee proteger.
6. Haga clic con el botón secundario del mouse en las páginas y, a continuación, seleccione Copiar.
7. Haga clic con el botón secundario en la nueva carpeta y, a continuación, seleccione Pegar.

Nota: si se han creado vínculos a estas páginas, éstos deben actualizarse para reflejar la nueva ubicación del contenido del sitio.

Para establecer permisos para el contenido Web

1. Haga clic en Inicio, Panel de control, Herramientas administrativas y, a continuación, haga doble clic en Administrador de Internet Information Server (IIS).
2. Haga clic con el botón secundario en la carpeta Sitios Web, el sitio Web, directorio, directorio virtual o archivo que desee configurar y seleccione Propiedades.
3. Active o desactive cualquiera de las siguientes casillas de verificación (si están disponibles), dependiendo del tipo de acceso que desee conceder o denegar:

Acceso al código fuente de secuencias de comandos. Los usuarios pueden tener acceso a los archivos de origen. Si Leer está activado, el origen se podrá leer; si se activa Escribir, se podrá escribir en el origen. La opción Acceso al código fuente de secuencias de comandos incluye el código fuente para las secuencias de comandos. Esta opción no se encuentra disponible si no se activan las casillas Leer o Escribir.

- **Leer** (activada de forma predeterminada). Los usuarios pueden ver las propiedades y el contenido del archivo o directorio.
- **Escribir.** Los usuarios pueden cambiar el contenido y las propiedades de un archivo o directorio.
- **Examen de directorios.** Se pueden ver las colecciones y listas de archivos.
- **Registrar visitas.** Se crea una entrada de registro para cada visita realizada al sitio Web.
- **Indizar este recurso.** Permite que Servicios de Index Server realice la indización de este recurso. Esto permite a los usuarios realizar búsquedas en el recurso.

4. En el cuadro de lista **Permisos de ejecución**, seleccione el nivel adecuado de ejecución de secuencia de comandos:

- **Ninguno.** No se ejecutan secuencias de comandos ni archivos ejecutables (por ejemplo, archivos del tipo .exe) en el servidor.
- **Sólo secuencias de comandos.** Se ejecutan únicamente secuencias de comandos en el servidor.
- **Secuencias de comandos y ejecutables.** En el servidor se ejecutan tanto secuencias de comandos como archivos ejecutables.

5. Haga clic en Aceptar. Si los nodos secundarios de un directorio disponen de permisos de sitio Web configurados, aparecerá el cuadro Omitir herencia.

6. Si este cuadro aparece, seleccione los nodos en la lista Nodos secundarios a la que desea aplicar los permisos Web del directorio.

O bien
Haga clic en Seleccionar todo para establecer la propiedad que se aplicará a los permisos Web en todos los nodos secundarios.
7. Si aparece más de un cuadro de opción Omitir herencia, seleccione los nodos secundarios en la lista del mismo nombre, o bien, haga clic en Seleccionar todo y, a continuación, en Aceptar para aplicar los permisos Web para esta propiedad a los nodos secundarios.
Si un nodo secundario que pertenece al directorio que dispone de permisos del sitio Web modificados también ha establecido los permisos para una opción en concreto, los permisos del nodo secundario anularán a los definidos para el directorio. Si desea que los permisos del sitio Web del nivel de directorio se apliquen a los nodos secundarios, debe seleccionar los nodos en el cuadro Omitir herencia.

Comprobación de la nueva configuración
Asegúrese de que se ha aplicado la configuración de seguridad adecuada al servidor Web.
Para comprobar que el acceso de escritura se deniega en los directorios de contenido del sitio Web

1. Presione Ctrl+Alt+Supr y, a continuación, haga clic en Cerrar sesión.
2. Inicie la sesión en el servidor Web utilizando una cuenta con permisos de lectura y ejecución en el directorio físico o virtual.
3. Haga clic en Inicio, haga clic con el botón secundario del mouse en Mi PC, seleccione Explorar y vaya a la ubicación del un archivo que desee copiar en el directorio físico o virtual.
4. Haga clic con el botón secundario del mouse en el archivo y, a continuación, seleccione Copiar.
5. Diríjase al directorio físico o virtual y, a continuación, haga clic con el botón secundario del mouse en el mismo. La selección de la función Pegar no está disponible en el menú contextual, lo que significa que no se dispone de acceso de escritura en el directorio.

Configuración del nivel de sockets seguro (Secure Sockets Layer, SSL) en el servidor Web

46. Configure características del nivel de sockets seguro (SSL) en el servidor Web para comprobar la integridad del contenido, la identidad de los usuarios y cifrar las transmisiones de red. La seguridad de SSL depende de un certificado de servidor que permite a los usuarios autenticar el sitio Web antes de que se transmita información personal como, por ejemplo, un número de tarjeta de crédito. Cada sitio Web puede disponer solamente de un certificado de servidor.

Obtención e instalación de certificados de servidor

Los certificados los envían organizaciones que no pertenecen a Microsoft y que se denominan entidades emisoras de certificados (CA). El certificado del servidor suele asociarse con el servidor Web, concretamente con el sitio donde se ha configurado SSL. Se debe generar una solicitud de certificado, enviarla a CA y, posteriormente, instalar el certificado tras ser recibido.

Para reforzar la seguridad, los certificados dependen de un par de claves de cifrado, una pública o otra privada. De hecho, al emitir la solicitud para un certificado de servidor, se está generando la clave privada. El certificado de servidor que se recibe de CA incluye la clave pública.

Requisitos

- **Credenciales:** la sesión se debe iniciar como miembro del grupo de administradores en el servidor Web.
- **Herramientas:** Administrador de Internet Information Server (Iis.msc) y el Asistente para certificados de servidor Web.

Para generar una solicitud para un certificado de servidor

1. Seleccione Inicio, haga clic con el botón secundario del mouse en Mi PC y, a continuación, haga clic en Administrar.
2. Haga doble clic en la sección Servicios y Aplicaciones y, a continuación, en Servicios de Internet Information Server.
3. Haga clic con el botón secundario del mouse en el sitio Web en el que desea instalar el certificado y, a continuación, seleccione Propiedades.
4. Haga clic en la ficha Seguridad de directorios. En la sección Comunicaciones seguras, haga clic en Certificado de servidor para iniciar el Asistente para certificados de servidor Web y, a continuación, seleccione Siguiente.
5. Haga clic en Crear un certificado nuevo y, a continuación, seleccione Siguiente.

6. Seleccione Preparar la petición ahora pero enviarla más tarde y, a continuación, Siguiente.

7. En el cuadro Nombre, indique un nombre que sea fácil de recordar. (El nombre predeterminado es el del sitio Web para el que se va a generar la solicitud de certificado como, por ejemplo, http://www.contoso.com.)

8. Especifique una longitud en bits y, a continuación, haga clic en Siguiente. La longitud en bits de la clave de cifrado determina la seguridad del cifrado. La mayor parte de las emisoras de certificados que no pertenecen a Microsoft prefieren que se seleccione un mínimo de 1024 bits.

9. En la sección Organización, indique la información de organización y unidad organizativa. Compruebe que la información es correcta y que los campos de organización no incluyen comas y, a continuación, haga clic en Siguiente.

10. En la sección Nombre común de su sitio Web, escriba el nombre del equipo host con el nombre de dominio y, a continuación, haga clic en Siguiente.

11. Indique la información geográfica y, a continuación, haga clic en Siguiente.

12. Guarde el archivo como .txt. (La ubicación y el nombre de archivo predeterminado son C:\certreq.txt.)

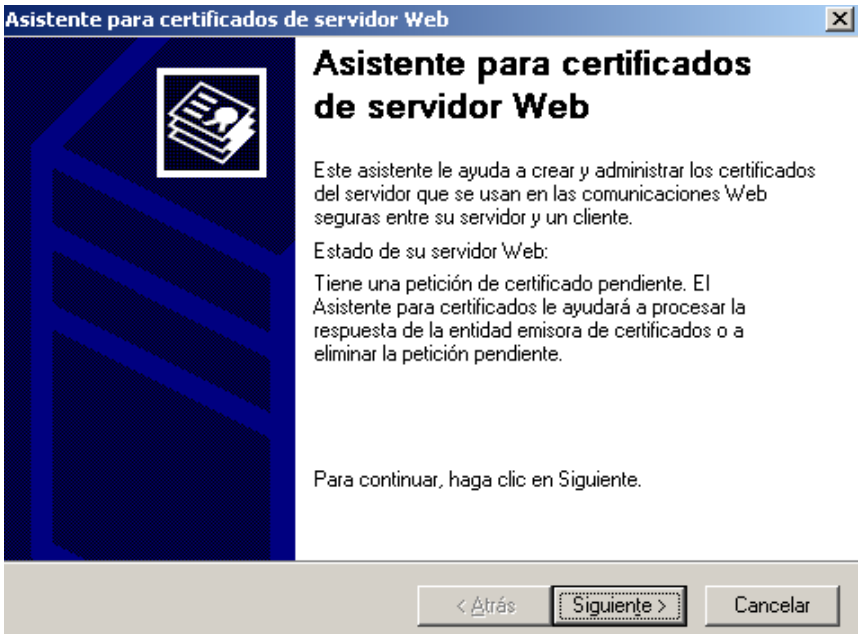

Figura 81 Asistente para certificado de servidor Web

En el siguiente ejemplo se muestra el aspecto de un archivo de solicitud de certificado.

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIDATCCAmoCAQAwbDEOMAwGA1UEAxMFcGxhbjgxDDAKBgNVBAsTA1BTUzESMBAG
A1UEChMJTWljcm9zb2Z0MRIwEAYDVQQHEwlDaGFybG90dGUxFzAVBgNVBAgTDk5v
cnRoIENhcm9saW5hMQswCQYDVQQGEwJVUzCBnzANBgkqhkiG9w0BAQEFAAOBjQAw
gYkCgYEAtW1koGfdt+EoJbKdxUZ+5vE7TF1ZuT+xaK9jEWHESfw11zoRKrHzHN0f
IASnwg3vZ0ACteQy5SiWmFaJeJ4k7YaKUb6chZXG3GqL4YiSKFaLpJX+YRiKMtmI
```

JzFzict5GVVGHsa1lY0BDYDO2XOAlstGlHCtENHOKpzdYdANRg0CAwEAAaCCAVMw
GgYKKwYBBAGCNw0CAzEMFgo1LjAuMjE5NS4yMDUGCisGAQQBgjcCAQ4xJzAlMA4G
A1UdDwEB/wQEAwIE8DATBgNVHSUEDDAKBggrBgEFBQcDATCB/QYKKwYBBAGCNw0C
AjGB7jCB6wIBAR5aAE0AaQBjAHIAbwBzAG8AZgB0ACAAUgBTAEEAIABTAEMAaABh
AG4AbgBlAGwAIABDAHIAeQBwAHQAbwBnAHIAYQBwAGgAaQBjACAAUAByAG8AdgBp
AGQAZQByA4GJAGKa0jzBn8fkxScrWsdnU2eUJOMUK5Ms87Q+fjP1/pWN3PJnH7x8
MBc5isFCjww6YnIjD8c3OfYfjkmWc048ZuGoH7ZoD6YNfv/SfAvQmr90eGmKOFFi
TD+hl1hM08gu2oxFU7mCvfTQ/2IbXP7KYFGEqaJ6wn0Z5yLOByPqblQZAAAAAAAA
AAAwDQYJKoZIhvcNAQEFBQADgYEAhpzNy+aMNHAmGUXQT6PKxWpaxDSjf4nBmo7o
MhfC7CIvR0McCQ+CBwuLzD+UJxl+kjgb+qwcOUkGX2PCZ7tOWzcXWNmn/4YHQl0M
GEXu0w67sVc2R9DlsHDNzeXLIOmjUl935qy1uoIR4V5C48YNsF4ejlgjeCFsbCoj
Jb9/2RM=
----END NEW CERTIFICATE REQUEST-----

Figura 82 Solicitud de certificado

13. Confirme los detalles de la solicitud, haga clic en Siguiente y, a continuación, en Finalizar.

Para enviar una solicitud para un certificado de servidor

1. Póngase en contacto con la entidad emisora para consultar los requisitos necesarios para enviar una solicitud.
2. Copie el contenido del archivo .txt creado en el anterior procedimiento en el formato que requiera su CA.
3. Envíe la solicitud.

Cuando reciba el certificado, podrá instalarlo en el servidor Web.

Para instalar un certificado de servidor

1. Copie el archivo de certificado (.cer) en C:\Windows\System32\CertLog.
2. Haga clic en Inicio, Panel de control, Herramientas administrativas y, a continuación, haga doble clic en Administrador de Internet Information Server (IIS).
3. Haga clic con el botón secundario del mouse en el sitio Web en el que desea instalar el certificado y, a continuación, seleccione Propiedades.
4. Haga clic en la ficha Seguridad de directorios. En la sección Comunicaciones seguras, haga clic en Certificado de servidor para iniciar el Asistente para certificados de servidor Web y, a continuación, seleccione Siguiente.
5. Haga clic en Procesar la petición pendiente e instalar el certificado y, a continuación, seleccione Siguiente.
6. Desplácese al certificado recibido. Haga clic dos veces en Siguiente y, a continuación, seleccione Finalizar.

Comprobación de la nueva configuración

Asegúrese de que se ha aplicado la configuración de seguridad adecuada al equipo local.

Para comprobar que un certificado se ha instalado en un servidor Web

1. Haga clic en Inicio, Panel de control, Herramientas administrativas y, a continuación, haga doble clic en Administrador de Internet Information Server (IIS).
2. Haga clic con el botón secundario del mouse en el sitio Web que incluye el certificado que desea ver y, a continuación, seleccione Propiedades.
3. En la sección Comunicaciones seguras de la ficha Seguridad de directorios, seleccione Ver certificado, revise el certificado y, a continuación, haga clic dos veces en Aceptar.

Aplicación y activación de conexiones SSL en el servidor Web

Tras instalar el certificado de servidor, se deben aplicar conexiones SSL en el servidor Web. A continuación, se deben habilitar las conexiones SSL.

Requisitos

- Credenciales: la sesión se debe iniciar como miembro del grupo de administradores en el servidor Web.
- Herramientas: Administrador de Internet Information Server (Iis.msc).

Para aplicar las conexiones SSL

1. Haga clic en Inicio, Panel de control, Herramientas administrativas y, a continuación, haga doble clic en Administrador de Internet Information Server (IIS).
2. Haga clic con el botón secundario del ratón en el sitio Web en el que desea aplicar las conexiones SSL y, a continuación, seleccione Propiedades.
3. Haga clic en la ficha Seguridad de directorios. En la sección Comunicaciones seguras, haga clic en Modificar.
4. Seleccione Requerir canal seguro (SSL), elija la seguridad de cifrado y haga clic en Aceptar.

Nota: si se especifica un cifrado de 128 bits, los equipos cliente que utilicen exploraciones de 40 ó 56 bits no se podrán comunicar con el sitio, a no ser que los exploradores se actualicen a versiones que admitan cifrado de 128 bits.

Para habilitar conexiones SSL en el servidor Web

1. Haga clic en Inicio, Panel de control, Herramientas administrativas y, a continuación, haga doble clic en Administrador de Internet Information Server (IIS).
2. Haga clic con el botón secundario del ratón en el sitio Web en el que desea habilitar las conexiones SSL y, a continuación, seleccione Propiedades.

3. Haga clic en la ficha Sitio Web. En la sección Identificación del sitio Web, compruebe que el cuadro Puerto SSL se completa con el valor numérico 443.
4. Haga clic en Avanzadas. Generalmente aparecen dos cuadros y la dirección IP y el puerto del sitio Web ya aparecen en el cuadro Varias identidades para este sitio Web. En el campo Varias identidades SSL para este sitio Web, haga clic en Agregar si no aparece el puerto 443. Seleccione la dirección IP del servidor, indique el valor numérico 443 en el cuadro Puerto SSL y, a continuación, haga clic en Aceptar.

Comprobación de la nueva configuración

Asegúrese de que se ha aplicado la configuración de seguridad adecuada al servidor Web.

Para comprobar conexiones SSL en el servidor Web

1. Abra el explorador e intente conectarse al servidor Web utilizando el protocolo estándar http://. Por ejemplo, en el cuadro Dirección, escriba lo siguiente: http://localhost

Si se fuerza SSL, aparecerá el siguiente mensaje de error:

Se debe ver la página mediante un canal seguro. La página a la que intenta tener acceso está asegurada con nivel de sockets seguro (SSL).

Intente de nuevo conectarse a la página que desee ver indicando lo siguiente: http://localhost

Generalmente aparece la página predeterminada del servidor Web. En nuestra aplicación, también se utilizó el Editor del Registro (Regedit.exe) para cambiar cuatro valores del registro con el fin de aumentar la seguridad. Todos ellos se recomiendan como prácticas más adecuadas, siempre que no se necesiten las funciones que se están desactivando.

4.2.5 Aceptabilidad

La aceptabilidad de un servidor Web, puede ser medida por varios aspectos, entre los cuales se encuentra uno muy importante, que es quienes van a decidir si el servidor es aceptable o no. En esta metodología se han definido tres parámetros para decir que un servidor Web Internet Information Server es aceptable. Estos parámetros son los siguientes:

Funcionamiento:

- Internet Information Server debe de visualizar páginas con máquinas que no se encuentren en la misma red. Esto nos garantizará que el servidor este funcionando correctamente.

- Configurar el registro de modo que muestre los logs, ya se diaria, semanal o mensualmente.
- Conexiones al sitio Web. Que estas conexiones estén acordes con el número de visitas que tiene el servidor Web. Ya que el poner un número excesivo de conexiones puede ocasionar que el servidor minimice su rendimiento.
- Habilitar el límite del ancho de banda. Para que cada sitio Web solo consuma el ancho de banda que se le asignado. En este caso para cada sitio se le signo un ancho de banda de 1024 Kb que es el ancho de banda necesario parta que funcione correctamente.
- Configuración del directorio particular: Configurarlo de tal modo que el directorio funcione de la mejor manera , esto se consigue haciendo las siguientes configuraciones:
    - Poner el directorio del equipo, o de un recurso compartido.
    - Permisos del sitio Web como son los de escritura, lectura, examinar directorios, registrar visitas.
    - Activar secuencia de comandos para ASP y ASP.NET .
    - Ejecuta inetinfo.exe en su propio proceso y ejecutar aplicaciones en sus propios procesos.
    - Establecer permisos secuencia de comandos y ejecutables.
    - Establecer asignaciones para la ejecución.
- Establecer asignaciones para la seguridad de directorios. En esto se ven los siguientes parámetros:
    - Restricciones de nombre de dominios y dirección IP, esto es muy importante porque nos dirá como se comportara nuestra intranet.

Rendimiento:

- Mejorar el rendimiento, deshabilitando servicios no esenciales como :
- Optimizar el rendimiento de datos para aplicaciones de red. Aquí se debe poner que la aplicación ocupe lo máximo para su rendimiento.
- Habilitar la supervisión de la CPU. Esta opción es para supervisar y apagar automáticamente los procesos de trabajo que consumen grandes cantidades de tiempo.
- Limitar las conexiones al sitio Web. Limita el número de conexiones para cada uno de los sitios. Esto depende mucho del número de visitas que tenga el sitio Web.
- Configuración de la Compresión HTTP .Esto se hace cuando el ancho de banda existe es escaso.

Seguridad

- Selección de componentes y servicios IIS fundamentales. Seleccionar únicamente aquellos que los sitios y aplicaciones Web necesitan para funcionar correctamente. Estos servicios son los siguientes:
- Habilitación de las extensiones de servicios Web esenciales: No habilitar todas las extensiones de servicios Web ya que el ámbito del ataque del servidor Web aumenta considerablemente, si necesita páginas dinámicas active Paginas Active Server.
- Deshabilitación de cuentas no utilizadas
    - Deshabilitación de la cuenta de invitado.
    - Cambio de cuenta de administrador.
    - Cambio de la cuenta IUSR_Nombre de equipo.
- Configuración de seguridad para archivos y directorios
    - Reubicación y definición de permisos para archivos de registro de IIS
    - Configuración de permisos de metabase de IIS.
- Seguridad en sitios Web y directorios virtuales.
    - Desplazamiento del contenido al sitio Web a una unidad que no es la del sistema.
    - Configuración de permisos del sitio Web.
- Configuración de permisos en directorios del sistema de archivos.,
- Configuración del nivel de sockets seguro (Secure Sockets Layer, SSL) en el servidor Web.

Cumplidas todas estas configuraciones en nuestro servidor Web, se ha a podido mejorar su funcionamiento, rendimiento y seguridad. Al haberse cumplido todos los parámetros podemos decir que nuestro servidor Web es aceptable.

CAPITULO V

APLICACIÓN DE LA GUÍA EN LOS SERVIDORES WEB

5.1APACHE

5.1.1 FUNCIONAMIENTO

1. Lo primero que se debe hacer es la descarga del servidor del servidor Apache de la página de internet http://www.apache .org, esta descarga se la realiza siempre que no se tenga el software del Servidor.
2. Unna ves que se halla descargado el softeare del servidor Apache se procede a la instalación como se indica en el punto dos de la guía, con esto se consigue una buena intalación de este servidor.

5.1.2 CONFIGURACION DE RENDIMIENTO Y SEGURIDAD

3. Antes de que se realice cualquier configuracion en el servidor se debe realizar una copiad de seguridad del Archivo httpd.conf lo que nos permitirá que proctejemos a nuestro servidor de cualquier error al momento de la configuración, además recomendamos se utilise el editor vi para toda modificación de las directivas del servido.
4. Con el punto dos de la guía para el rendimiento y la seguridad, establecemos que todos lo archivos de nuestro servidor se almacenen en una ruta específica con lo que ganamos mayor orden en la ubicación de los archivos que el servidor va a utilizar.
5. Con la directiva PidFile logramos determinar la ubicación de los Id de procesos de cada una de las tareas que el servidor realiza los mismo que se deben guardar en el archivo httpd.pid.
6. Para lograr que las peticiones se las realice atravéz de un determinado puerto o dirección IP lo hacemos con la directiva listen. Con esto logramos que el servidor pueda escuchar en uno o varias direciones IP dependiendo de su tamaño.

7. Con DocumentRoot especificamos la ruta donde Apache tiene que buscar los archivos para entregarle al cliente, ganado asi que más orden y un mejor control.

8. Cuando paparecen las paginas de error siempre es bueno que el usuario sepa a donde o a quien se le debe preguntar haceca del error, para ello la directiva Server Admin nos permite especificar el Email del administrador del sistema.

9. De iguial manera que el anterior, nuestrio servidor debe tener un nombre y para poder acceder a el, para eso debemos especificarlo con la directiva ServerName.

10. Lo siguiente es configurar la *TimeOut* la misma que se encarga de establecer el tiempo de caducidad de una conexión, y su valor depende del siguiente calculo:

μ=Es la taza de llegada
λ=Es la taza de servicio
μ=0.50m
λ=0.05m

La formula a aplicar son formulas de la teoría de colas y la formula para este caso es:

$W=1/(\mu-\lambda)$.
$W=1/(0.50-0.05)=120s$

Este valor de 120 segundos es el óptimo para que una vez cumplido se cierre la conexión

11. Cuando el cliente realiza una petición a una página este debe realizar una solicitud por cada objeto y esto significa establecer una conexión para cada solicitud. Actualmete nuestro servidor se encuentra de la siguiente manera: *KeepAlive* off lo que ocaciona es lo siguiente:

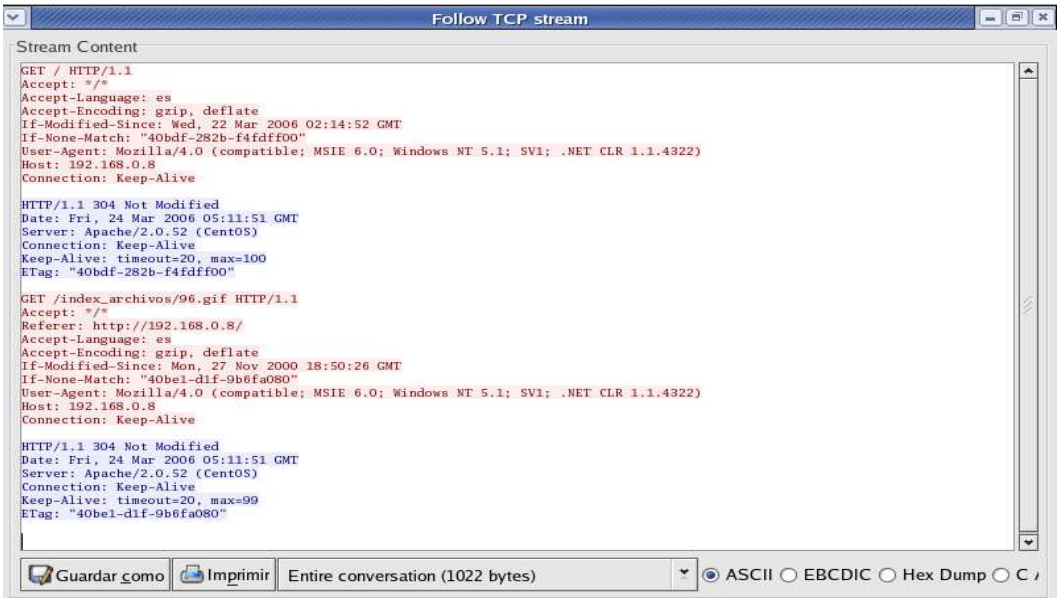

Figura 83: Podemos ver el contenido de uno de los procesos TCP

Vemos el número de proceso que se crean cuando esta directiva está en off. Veamos que ocurre cuando esta directiva la configuramos con el valor de On:

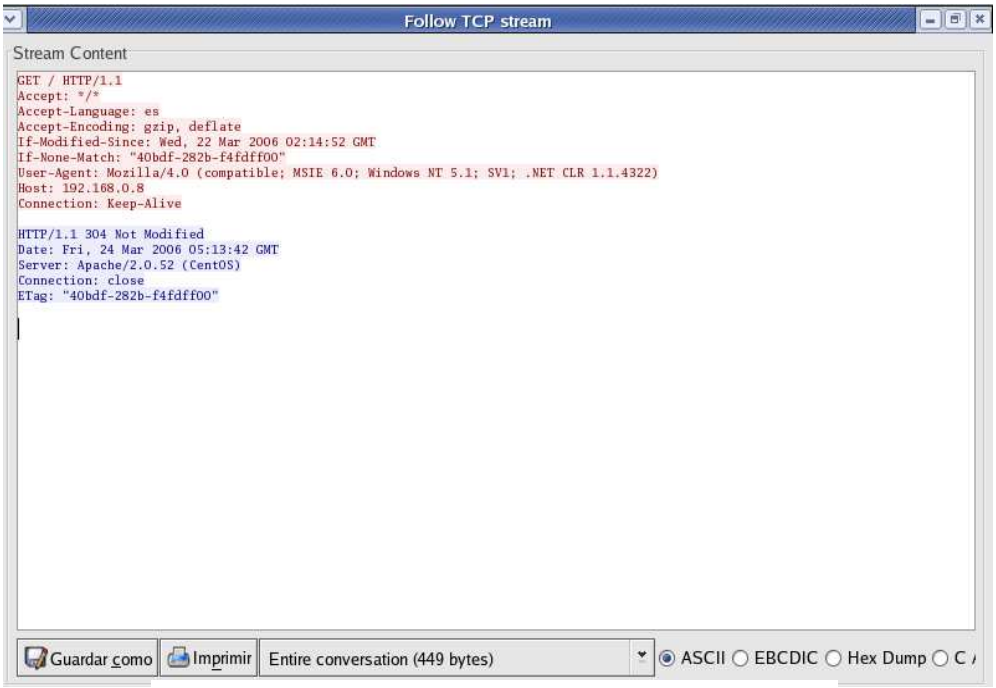

Figura 84: Proceso TCP cuando *KeepAlive* se encuentra en *off*

Vemos que el número de procesos disminuyó debido a que ya no se realiza una conexión para cada solicitud sino que en una conexión se atienda a varias

12. Una vez que hemos configurado para que varias solicitudes sean atendidas por una sola conexión es necesario que le digamos a Apache cuantas son la solicitudes que se deben atender en dicha conexión y esto lo hacemos de la siguiente manera:

Con los valores anteriores:

$L=\lambda^*\lambda/\mu(\mu-\lambda)$

Quedando:

$L=0.25/0.0225=33s$

Entonces:

MaxKeepAliveRequests 33

Que sería el valor optimo para este servidor, ganando de esta forma que la conexión no tenga que esperar a los 120s para cerrarce sino que lo puede hacer cuando haya completado las 33 solicitudes.

13. Podemos setewra el tiempo entre peticiones para que un cliente, con esto ganamos que el servidor no tenga que esperar mucho paraedsperar una nueva petición, esto se logra con la directiva KeepAliveTimeOut

14. Cuando el servidor realiza procesos de realización de nombres de dominio hace que el servidor sea demaciado lento comparado con el caso de que no lo hiciera. Para evitar que el servidor haga este proceso, lo hemos configurado de la siguiente forma:

UseCanonicalName off

15. Una directiva muy importante es DirectoryIndex que nos permite dedirle a nuetro servidor que tipos de páginas debe ejecutar primero. Esta directiva es DirectoryIndex.

16. Siempre que algún usuario vista nuestra página nuestro servidor debe registrarlo en el archivo de accesos quien fu a visitarnos para ello con la directiva HostNameLookUps podemos dedirle a nuestro servidor que registre el nombre del visitante.

17. Para evitar que en las páginas de error aparezca información de nuestro servidor lo hemos hecho con la siguiente directiva.

ServerAsignature Off

18. No solo de las paginas de error la información de nuestro servidor es importante sino tambien de las cabeceras de los paquetes que se envian por ello hemos puesto la siguiente directiva en nuestro servidor de la siguiente forma:

ServerToken SO

19. Hemos especificado un numero máximo y minimo de procesos inactivos y lo hemos hecho de la siguiente manera:

MinSpareServers 15
MaxSpareServers 33

Con esto logramos que los procesos inactivos esten dentro de los valores consernienmtes en nuestro servidor y así no ocupar procesador ni memoria.

20. Según los calculos hechos anteriormente nuetro servidor debe tener un máximo de 33 procesos en cada conexión para que su rendimiento sea el optimo, y de la misma manera en los procesos secundarios:

MaxRequestsPerChild 33

21. Una vez que ya tenemos el número de solicitudes para ser atendidad ese mismo número es igual a lnúmero de clientes que vamos atender, esto lo hacemos con la directiva MaxClients.

22. Si ya sabemos que nuestro servidor va atender 33 solicitudes y a 33 clientes entonses lo maximo de clientes que podemos tener en cola son 33, lo cual lo especificamos con ListenBacklog.

Lo siguiente a tyener encuenta es la seguridad, la configuración ssl, con la misma ganamos es la seguridad de la transmisión del paquete atraves de encriptación y de fidmas digitales.

5.2 IIS (INTERNET INFORMATION SERVER 6.0)

5.2.1 FUNCIONAMIENTO

1.-Este punto de la guia no se aplica para el caso del servidor de la ESPOCH porque el IIS ya se encuentra instalado.

2.- Este punto de la guia no se aplica para el caso del servidor de la ESPOCH porque el IIS ya se encuentra instalado.

3.-Este punto de la guia no se aplica para el caso del servidor de la ESPOCH porque el IIS ya se encuentra instalado.

4.-Este punto de la guia no se aplica para el caso del servidor de la ESPOCH porque el IIS ya se encuentra instalado.

5.-Este punto de la guia no se aplica para el caso del servidor de la ESPOCH porque el IIS ya se encuentra instalado.

6.-Este punto de la guia no se aplica para el caso del servidor de la ESPOCH porque el IIS ya se encuentra instalado.

7.-Este punto de la guia no se aplica para el caso del servidor de la ESPOCH porque el IIS ya se encuentra instalado.

8.-Este punto de la guia no se aplica para el caso del servidor de la ESPOCH porque el IIS ya se encuentra instalado.

9.-Este punto de la guia no se aplica para el caso del servidor de la ESPOCH porque el IIS ya se encuentra instalado.

10.-Este punto de la guia no se aplica para el caso del servidor de la ESPOCH porque el IIS ya se encuentra instalado.

11.-Este punto de la guia no se aplica para el caso del servidor de la ESPOCH porque el IIS ya se encuentra instalado.

5.2.2 OPCIONES DE CONFIGURACION

12.- Se debe de habilitar el registro diriamente como lo muestra a continuación:

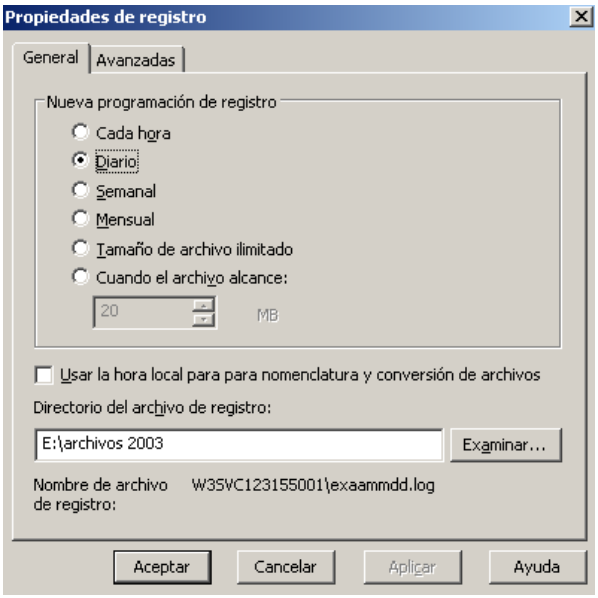

Figura 85 Propiedades de registro

Para que halla un mejor registro de las visitas, ya que esto es muy importante para calculos que inciden en el rendimiento del servidor web.

13.- Este punto de la guia no se aplicaen el servidor de la ESPOCH porque ya existen mensajes de error personalizados.

14.-Este punto de la guia no se aplica en el servidor de la ESPOCH porque estas opciones ya salen por defecto.

15.- Este punto de la guia no se aplica en el servidor de la ESPOCH porque estas opciones ya salen por defecto

16.- Este punto de la guia no se aplica en el servidor de la ESPOCH porque estas opciones ya salen por defecto

17.- Este punto de la guia no se aplica en el servidor de la ESPOCH porque estas opciones ya salen por defecto

18.- Este punto de la guia no se aplica en el servidor de la ESPOCH porque estas opciones ya salen por defecto

19.- Este punto de la guia no se aplica en el servidor de la ESPOCH porque estas opciones ya salen por defecto

20.- Este punto de la guia no se aplica en el servidor de la ESPOCH porque estas opciones ya salen por defecto

21.- Este punto de la guia no se aplica en el servidor de la ESPOCH porque estas opciones ya salen por defecto

22.- Este punto de la guia no se aplica en el servidor de la ESPOCH porque estas opciones ya salen por defecto

23.- Este punto de la guia no se aplica en el servidor de la ESPOCH porque estas opciones ya salen por defecto

24.- Este punto de la guia no se aplica en el servidor de la ESPOCH porque esta opcion es muy delicada ya que se tiene que saber a la perfeccion que IPs no se admitir´ña para que ingresen al sitio web.Configurar mal esta opción podría hacer que el servidor deniegue el acceso a todas las IPs que deseen conectarse a un sitio.

25.- Este punto de la guia no se aplica en el servidor de la ESPOCH porque estas opciones ya salen por defecto.

5.2.3 OPTIMIZACION

Deshabilitación de servicios no esenciales:

26. Los servicios que deben ser deshabilitados son los siguientes:

- Servicio de alerta
- Portafolios
- Examinador de equipos
- Cliente DHCP
- Servidor DHCP
- Servicio de fax
- Replicación de archivos

- Monitor de infrarrojos
- Conexión compartida a Internet
- Messenger
- Escritorio remoto compartido de NetMeeting
- DDE de red
- DSDM de DDE de red
- NetBIOS de NWLink
- NWLink IPX/SPX
- Cola de impresión
- Servicio de ayuda TCP/IP NetBIOS
- Telefonía
- Telnet
- Sistema de alimentación ininterrumpida

27.- Este punto de la guia no se aplica en el servidor de la ESPOCH porque estas opciones ya salen por defecto

28.- Este punto de la guia no se aplica en el servidor de la ESPOCH porque estas opciones ya salen por defecto.

Habilitar el límite de Ancho de Banda

29. Para el cálculo del ancho de banda se usan las siguientes formúlas:

- 5 visitas el día de mayor tráfico.
- 12 páginas por visita.
- 100 KB por página.

5 x 12 x 1000 x 1, 25 = 75.000 KB Transferencia Diaria

Ahora lo dividiremos por horas, pero hay que tener presente que no a todas horas hay el mismo tráfico. La hora punta es 1,6 veces superior a la media horaria. De este modo, el volumen de transferencia en hora punta es de:

75.000 KB / 24 x 1, 6 = 5.000 KB

Se divide para 24 por que 24 son las horas del día.

Para complicarnos el cálculo, tampoco entran al mismo ritmo todos los visitantes, hay momentos en que coinciden varias conexiones de visitantes, otros en los que no coincide ninguno. Si queremos cubrir estas oscilaciones, nuevamente tendremos que aplicar un factor de corrección. En él decidimos las posibilidades de que haya más o menos tiempo de lentitud (atascos de tráfico) en las respuestas en los peores momentos. Por fortuna, otro de los inconvenientes está en que el ancho de banda no

se contrata en las unidades exactas que queramos, así que bastará con contratar el valor inmediatamente superior al que obtendremos en la media de la hora punta, siempre que deje un poco de margen.

Teníamos un valor en Kilobytes (KB) promedio por hora, y hemos de convertirlo a Kilobits por segundo (Kbps). El último cálculo ya es el siguiente:

$$5.000 \text{ KB} / 60 / 60 \times 8 = 11.11 \text{ Kbps}$$

Siendo los valores de ancho de banda contratables de 128Kbps, 256Kbps, 512Kbps, etc., nos decidiríamos por el inferior, 128Kbps. En el caso de ejemplo usado, el volumen de transferencia de datos nunca nos llevará a una contratación por ancho de banda, pues no necesitamos tanta comunicación.

Pero como la unidad en la que se debe poner el ancho de banda en el sitio es en kylobytes, el ancho que deberia de haber para el sitio del sistema financiero es de 5000 KB.

Habilitar la supervisión de la CPU

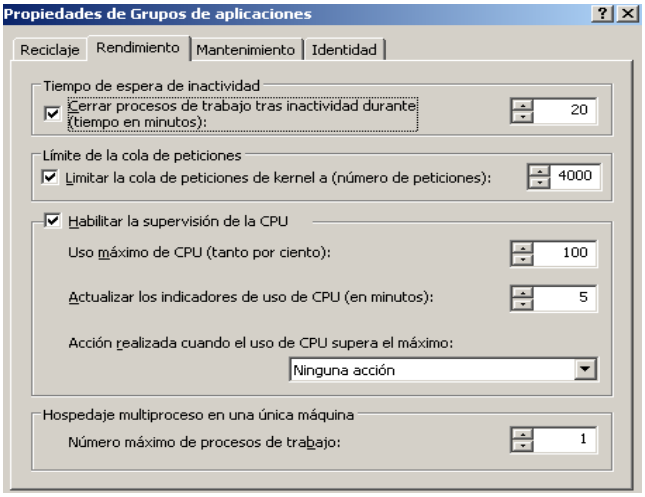

Figura 86 Habilitar supervisión CPU

30. Los valores que deben de ir es cuando el proceso ocupe el 100% de uso de CPU se apague, y que esto se monitoree cada 5 minutos.

Límites de conexiones al sitio web

31. Estos parámetros deben ser calculados de acuerdo al tráfico que tenga el sitio web.

En el caso del sitio web del Sistema Financiero de la ESPOCH, al revisar sus archivos logs se arrojarón los siguientes datos:

Del periodo del 10 de mayo al 31 de Julio del 2006 se tuvo un total de 136 visitas únicas es decir, solo se verifica que el usuario visito el sitio.

En esas 136 visitas, se hicierón un total de 1024 conexiones simultáneas. Si dividimos la cantida de 1024 para 136 nos sale un promedio de 7 conexiones por usuario. Si eso dividimos las 136 visitas unicas que se tuvo en el periodo para el promedio de 7 conexiones por usuario nos saldra que 19,42 usuarios podran conectarse simultáneamente sin tener inconvenientes.Pero se recomienda que de este valor se eleve un poco por eso se pondra un valor de 200.Esto es para las conexiones al sitio.

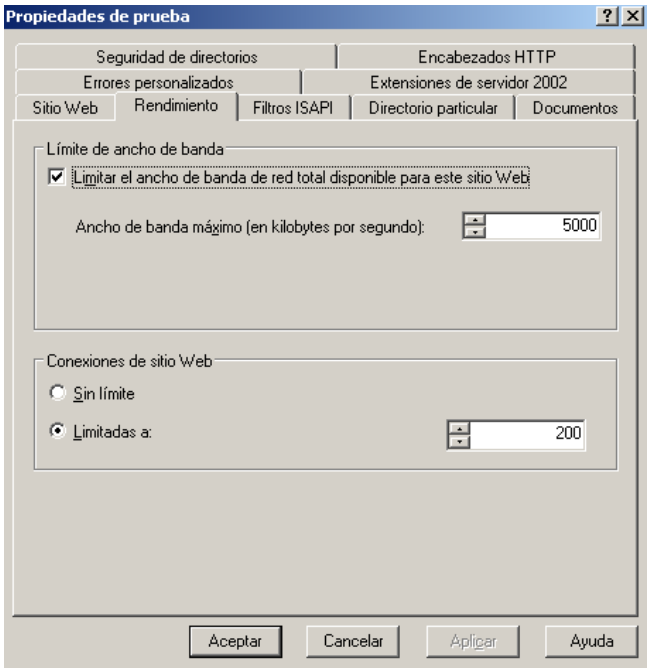

Figura 87 Limitar ancho de banda y conexiones al sitio

32.- Este punto de la guia no se aplica en el servidor de la ESPOCH porque estas opciones ya salen por defecto

Compresion Http

33. Para esto se hace lo siguiente:

34. Agregar la extensión de servicio web

35. Habilitar la compresión estática y dinámica

36. Hacer copia de seguridad de la metabase

Poner los sigientes valores en la metabase:

HcdoStaticCompression=”TRUE”

HcDynamicCompressionLevel=9

HcFileExtensions=”htm html txt”

HcOnDemandCompLevel=”9”

HcPriority=”1”

HcScriptsFileExtensions=”asp aspx dll js exe”

HcCacheControlHeader=”max-age=86400”

HcCompresionBufferSize=”8192”

HcDoDiskSpaceLimiting=”False”

5.2.4 SEGURIDAD

37.- Selección de componentes y servicios de IIS fundamentales

Subcomponente o servicio	**Configuración predeterminada**	**Configuración del servidor Web**
Extensiones de Servicio de transferencia inteligente en segundo plano (BITS)	Deshabilitado	Sin cambio
Archivos comunes	Habilitado	Sin cambio
Servicio FTP	Deshabilitado	Sin cambio
Extensiones de servidor de FrontPage 2002	Deshabilitado	Sin cambio
Administrador de servicios de Internet Information Server	Habilitado	Sin cambio
Impresión de Internet	Deshabilitado	Sin cambio
Servicio NNTP	Deshabilitado	Sin cambio
Servicio SMTP	Habilitado	Deshabilitado
Servicio World Wide Web	Habilitado	Sin cambio

Habilitación de las extensiones de servicios Web esenciales

38.-Esto se refiere a las extensiones que deben de ser habilitadas, en todo servidor solo las Active Page Server para ASP.net y las Extensiones de Servidor FrontPage

Deshabilitación de cuentas no utilizadas

39.-Deshabilitación de la cuenta de invitado

Figura 88 Deshabilitacion Cuenta invitado

40.-Cambio de nombre de la cuenta de administrador

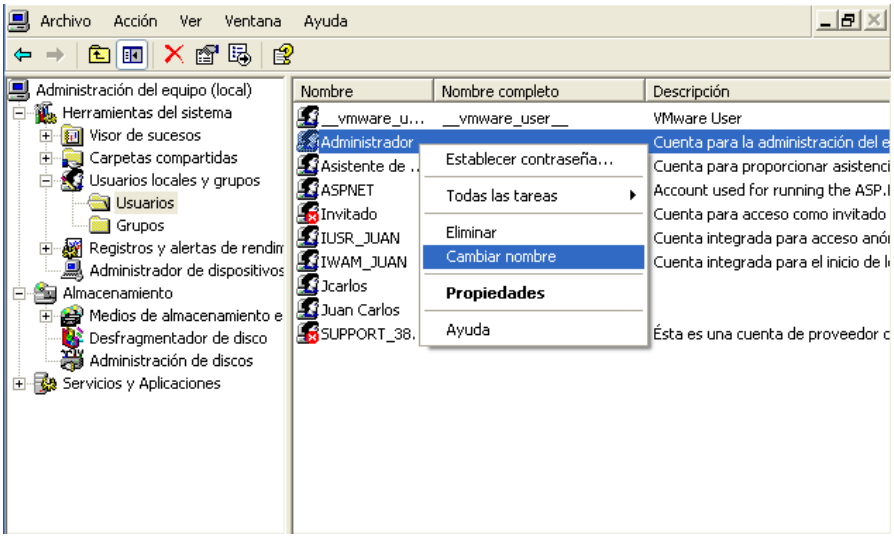

Figura 89 Cambio de nombre de la cuenta administrador

41.-Cambio de nombre de la cuenta IUSR

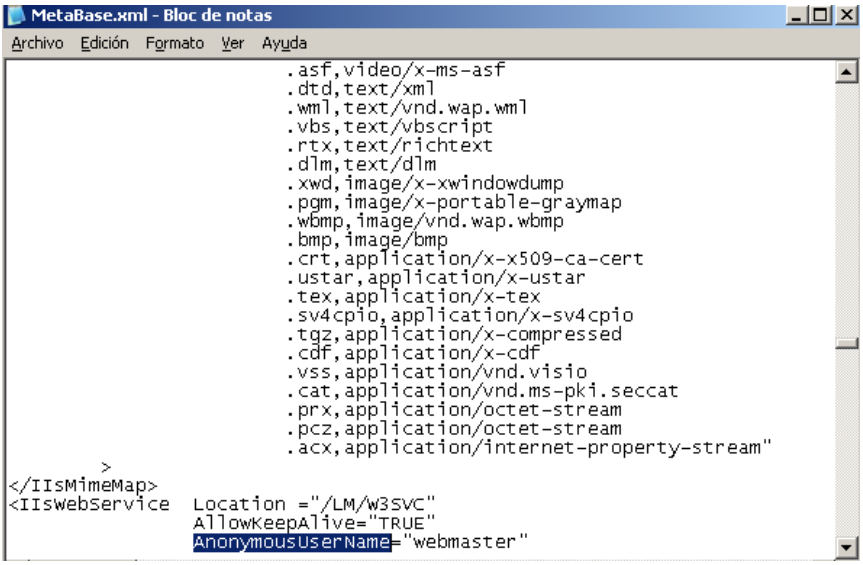

Figura 90 Propiedades de registro

42.- Este punto de la guia no se aplica en el servidor web de la ESPOCH porque este punto ya estaba implantado.

43.-Configuración de permisos de metabase de IIS.

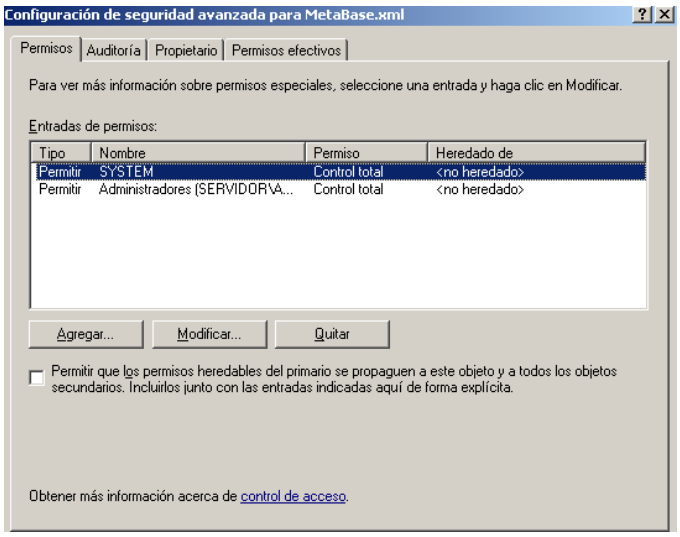

Figura 91 Configuracion permisos metabase

44.- Este punto de la guia no se aplica en el servidor web de la ESPOCH porque este punto ya estaba implantado.

45.-Configuración de permisos del sitio Web.

46.- Configuración del nivel de sockets seguro

-----BEGIN NEW CERTIFICATE REQUEST-----

MIIDATCCAmoCAQAwbDEOMAwGA1UEAxMFcGxhbjgxDDAKBgNVBAsTA1BTUzESMBAG
A1UEChMJTWljcm9zb2Z0MRIwEAYDVQQHEwlDaGFybG90dGUxFzAVBgNVBAgTDk5v
cnRoIENhcm9saW5hMQswCQYDVQQGEwJVUzCBnzANBgkqhkiG9w0BAQEFAAOBjQAw
gYkCgYEAtW1koGfdt+EoJbKdxUZ+5vE7TF1ZuT+xaK9jEWHESfw11zoRKrHzHN0f
IASnwg3vZ0ACteQy5SiWmFaJeJ4k7YaKUb6chZXG3GqL4YiSKFaLpJX+YRiKMtmI

JzFzict5GVVGHsa1lY0BDYDO2XOAlstGlHCtENHOKpzdYdANRg0CAwEAAaCCAVMw
GgYKKwYBBAGCNw0CAzEMFgo1LjAuMjE5NS4yMDUGCisGAQQBgjcCAQ4xJzAlMA4G
A1UdDwEB/wQEAwIE8DATBgNVHSUEDDAKBggrBgEFBQcDATCB/QYKKwYBBAGCNw0C
AjGB7jCB6wIBAR5aAE0AaQBjAHIAbwBzAG8AZgB0ACAAUgBTAEEAIABTAEMAaABh
AG4AbgBlAGwAIABDAHIAeQBwAHQAbwBnAHIAYQBwAGgAaQBjACAAUABvAG8AdgBp
AGQAZQByA4GJAGKa0jzBn8fkxScrWsdnU2eUJOMUK5Ms87Q+fjP1/pWN3PJnH7x8
MBc5isFCjww6YnIjD8c3OfYfjkmWc048ZuGoH7ZoD6YNfv/SfAvQmr90eGmKOFFi
TD+hl1hM08gu2oxFU7mCvfTQ/2IbXP7KYFGEqaJ6wn0Z5yLOByPqblQZAAAAAAAA
AAAwDQYJKoZIhvcNAQEFBQADgYEAhpzNy+aMNHAmGUXQT6PKxWpaxDSjf4nBmo7o
MhfC7CIvR0McCQ+CBwuLzD+UJxl+kjgb+qwcOUkGX2PCZ7tOWzcXWNmn/4YHQl0M
GEXu0w67sVc2R9DlsHDNzeXLIOmjUl935qy1uoIR4V5C48YNsF4ejlgjeCFsbCoj
Jb9/2RM=
----END NEW CERTIFICATE REQUEST-----

Figura 82 Solicitud de certificado

CONCLUSIONES

- La compresión http es muy beneficiosa sobre todo en los sitios en los que no se goza de mucho ancho de banda del lado del servidor o del lado del cliente.
- Son muy pocas las configuraciones, que el Servidor Web IIS tiene en si como servidor, ya que la mayoria de las configuraciones se realizan en los sitios Web, aunque algunas de estas tales como el limite de conexiones y el limite de ancho de banda del sitio mejoran el redimiento del servidor Web.
- El archivo de la metabase es muy parecido a lo que es el httpd.conf en Linux, en este archivo se pueden hacer muchos cambios en caliente, es decir mientras el IIS este corriendo, esta es una de las ventajas de las que goza este servidor Web.
- El análisis de visitas de un servidor web nos permite saber cuanto tráfico recibe un servidor, convirtiendose esto en un factor muy importante a la hora de optimizar el rendimiento, y seguridad de un servidor web.
- Para llegar a los valores óptimos de cada una de las directivas hemos utilizado calculos matemáticos de la teoria de colas.
- Los valores que Apache le da a cad una de usu directivas al momento de la instalación forman un punto de equilibrio en el rendimiento de cualquier servidor sin importar su concurrencia.

RECOMENDACIONES

- Como recomedacion quisieramos, decir que cuando se configure el archivo de la metabase en el IIS, se lo abra con el bloc de notas y no con con el wordpad ya que este formato hace cambiar el contenido del archivo de la metabase lo que ocasiona que el IIS no funcione correctamente.
- Antes de hacer cualquier configuración en el servidor se deberá sacar una copia ya sean de archivos, o carpetas que tengan que ver con las configuraciones que se vayan a realizar.
- Se debe tener mucho cuidado con el archivo httpd.conf por lo que se recomienda sacar una copia de este, ya que si se efectua alguna configuración incorrecta en este archivo, se dañaria el correcto funcionamiento del servidor web apache, lo que ocasionaria que habria que volverlo a instalar.
- Para poder visualizar cuanto se ha mejorado el rendimiento del servidor web, se recomienda utilizar herramientas como el monitor del sistema y el ethereal, estas nos permiten visualizar el trafico de bytes, paquetes, uso de memoria, uso del procesador lo que nos permite verificar cuanto se ha mejorado el rendimiento de un servidor.
- La compresión http no se debe de usar en servidores donde hay mucho uso de procesador, ya que está consume tambien tiempo de procesador y esto podría ocasionar que el servidor no aumente su rendimiento, más bien este rendieminto se vea afectado por la compresión.
- Al poner un número de conexiones al sitio web, lo óptimo seria que se pusiera un limite muy cercano a la cantidad de usuarios que lo visitan, ya que de lo contrario esto haria que el servidor ocupára mucha cantidad de memoria.
- Apache fue diseñado para ser primero seguro y luego correcto, es por eso que sus directivas se basan en rendimiento porque este parámetro es el que varía depoendiendo de las condiciones y de las concurrencias.
- Es necesario que se realice una revisión periódica de las visitas de nuestro servidor debido a que estas pueden aumentar o disminuir y sis esto ocurre los valores de las directivas deben cambiar.

RESUMEN

Mejorar el rendimiento, seguridad, funcionamiento y aceptabilidad de un servidor web a traves de la elaboración y aplicación de una guia referencial.

Para esto se uso:

Programas como el analizador de visitas awstats, el monitor del sistema de windows 2003 y el monitor del sistema del White Box, dos máquinas pentium cuatro, un patch cord, dos tarjetas de red. Está guia se realizó usando la metodología de analisis de registros que permite trabajar con ellos y verificar los usuarios del servidor, la herramienta awstats que permitió analizar las visitas diarias de los usuarios, el monitor del sistema mostró el uso de la CPU, memoria y bytes transferidos.

Con esto se obtuvo la elaboración de una guia, fácil de aplicar, y segura porque cada paso de está fue comprobado. Al aplicar está guia se obtuvo los siguientes resultados:

El rendimiento mejoró un 0,5% gracias a las directivas de configuración KeepAlive, TimeOut,MaxKeepAliveRequest, KeepAliveTimeOut y la compresión http. El funcionamiento aumentó un 10% al configurar su instalación y los permisos de ejecución del sitio web, la seguridad se incrementó un 15% al deshabilitar cuentas de usuarios invitados, y servicios como el STMP, ya que estos ocasionan que el servidor sea inseguro.La aceptabilidad creció un 20% ya que el servidor con los parámetros anteriores se hizó más rápido, fiable y seguro.

Con la aplicación de esta guía se consiguó mejorar el rendimiento, funcionamiento, seguridad y aceptabilidad del servidor web.

Se recomienda usar esta guía para mejorar el rendimiento, funcionamiento, seguridad y aceptabilidad de un servidor web

SUMMARY

The Primary target of our thesis was to develop a referential guide to diagnose and to improve the yield, security, accessibility, acceptability and operation through the analysis and technological optimization of the Web servers of the ESPOCH.

When making our thesis we used the following methods and metodologias that us sirvierón to be able to orient to us of one better way: Inductive Method: Method to prove technological and particular concepts applied to a general.

Method of Heuristic evaluation analyzes the requirements and makes studies. Method Colike. It calculates the effort and cost of a project. Methodology of Analysis of Registries. It analyzes the registries of the Web servers, teaches as to treat to these registries and as to work with them and it verifies that people visit the Web server.

On the basis of all the methods and methodology before mentioned it was created an own methodology that contain techniques and methods that allow to optimize the yield, security, operation and acceptability of a Web server.

The yield of the servant in a 0.5% improved this verified it to traves of tools that allow to observe the transferred use of memory, CPU, and bytes and of tools that allow to see the traffic us of visits. It was obtained that the Web server will work of better way when making to him some configurations that improve his funcionamiento. The security was increased when deshabilitar accounts, services and components that cause that a Web server uncertain and is ready to attacks El much more acceptable Web server hizó to the user, since became a faster Web server, trustworthy and safe.

GLOSARIO

- **APACHE:** Servidor Web de Linux.
- **AVPKT:** Tamaño medio del paquete medido en bytes
- **AWSTATS:** Analizador de visitas, analizador de ficheros logs.
- **BANDWITH:** Ancho de banda
- **CBQ**: Disciplina de Colas que permite regular el ancho de banda.
- **HTTPD.CONF**: Ficehro de configuracion del apache
- **HTTPS**:protocolo ustilzado exclusivamente en comunicaciones de hipertexto
- **IIS:** Servidor Web de Windows.
- **METABASE:** Archivo de configuracion de IIS.
- **MPU:** Tamaño minimo del paquete.
- **NCSA:** Formato de archivo de log de IIS.
- **REGISTRO DE TRANSFERENCIA:** Donde se registran todos los accesos al servidor
- **RATE:** Ancho de Banda regulado con el que queremos que funcione nuestra disciplina de colas
- **SSL**: Secure Soket Layer (Capa de Socket Seguro).
- **SSH**: Protocolo usado exclusivamente en reemplazo de telnet.
- **SFQ**:Algoritmo de ancho de banda que busca equidad para todas las peticiones
- **SQUID**: Servidor proxy de Linux
- **W3SC:** Formato de archivo log de IIS.
- **WSRM**:Servicio de Windows

BIBLIOGRAFIA

1. GIL, Francisco Javier. Creación de Sitios Web con PHP 2da.ed Barcelona: PrenticeHall, 1998 pp. 160-190
2. MARTINEZ, José Andrés. Linux Referencia Visual 4ta.ed Buenos Aires:McGrawHill,2000 pp 10-85
3. BOWEN, Rich .Servidor Apache al descubierto 2da.ed Buenos Aires:McGrawHill,2000 pp 100-150
4. RICK/STOUT. Optimización de Servidores Web Análisis y Estadística 1ra.ed Madrid:PrenticeHall,2000 pp 15-56,80-190

BIBLIOGRAFIA INTERNET

1. APACHE

http://www.Apache.org/httpd.html

(2006/05/20)

http://www.Apache.org/info/known_bugs.html

(2006/06/20)

http://www.Apache.org/

(2006/07/20)

http://www.apachecon.com/

(2006/08/20)

http://www.redhat.org/apache.htm/

(2006/08/02)

http://www.guarh.up.pt/apacheES/manual-es/misc/perf-tunning.html

2. AWSTATS

http://www.sourceforgenet.com/awstas/awstats.exe

(2006/07/16)

http://www.osmoniasis.com

(2006/09/24)

http://www.softdownload.com

(2006/11/24)

3. IIS 6.0

http://www.msdnlatam.com/iis 6.0/iis.htm

(2006/11/24)

http://www.microsoft.com/servidor.htm

(2006/11/24)

http://www.ilustrados.com/administracion/iis.htm

(2006/11/24)

http://www.Albergueweb.com/arquitectura/servicios.htm

(2006/11/24)

http://www.rackhispano.com/servidore.html

(2006/11/24)

http://www.virtualpyme.com/h10.html

(2006/11/24)

http://www.slackwere.cl/gua/arreglo/cap.html

(2006/11/24)
http://www.desarrolloweb.com/articulos/registros_iis.htm
(2006/11/24)
http://www.mexcompany.com/apuntes/administracion_iis_6.0.htm
(2006/11/24)
http://www.islaweb.com/tecnicos/granjaweb.htm
(2006/11/24)
http://www.bandaancha.com/qos/iis_art.htm
(2006/11/24)

4. METODOLOGIA DE ANALISIS DE REGISTROS

http://www.informatizate.net/articulos/metodologias_de_desarrollo_desoftware_07062004.html

(2005/08/09)

http://www.angelfire.com/scifi/jzavalar/apuntes/Metodologia.html

(2005/08/09)

http://www.clikear.com/manuales/rom/rom3.asp

(2005/08/09)

http://www.lug.fi.uba.ar/documentos/lista-herramientas/

(2005/08/03)

http://www.monografias.com/trabajos12/docmento/docmento.shtml

(2005/08/03)

http://galeon.hispavista.com/betovilche/is.htm

(2005/08/03)

5.-PERL

http://www.activeperl.com/download active/active.htm
(2006/09/16)

ANEXOS

ANEXO 1

HERRAMIENTAS UTILIZADAS PARA EL ANALISIS DE VISITAS

AWSTATS

PASOS PARA CONFIGURAR EL AWSTATS EN WINDOWS 2003.

Si usted usa un paquete proporcionado con una distribución de Linux o instalador de windows, como indica el paso 0-1 usted no lo conoce, hágalo de nuevo.

Después de transmitir y extraer el paquete awstats, usted debe ejecutar el archivo awstats _configure.pl para hacer algunos acciones de instalación. Ud encontrará en el awstats el directorio (si usando el instalador de windows, la escritura se lanza automáticamente): el perl awstats_configure.pl

Esto eso que el hacer o preguntar (usted puede hacer todos esos pasos por mano en lugar de awstats_configure.pl corriente si usted prefiere);

Con el Internet Información Servidor

Paso 0-1

Configure IIS para anotar en W3C Extendido anote el formato" (Usted todavía puede usar su propio formato del leño pero el arreglo es más fácil si hecho como sugerido). Para que, para esto, empiece el IIS Chasquido-en, seleccione el sitio web y mire sus Propiedades. Escoja W3C Extended el Leño Estructure, entonces las Propiedades, entonces la Etiqueta Extendió Propiedades y uncheck todo bajo las Propiedades Extendidas. Una vez ellos es todo desenfrenado, verifique todos los campos del partidario:

date

time
c-ip
cs-username
cs-method
cs-uri-stem
cs-uri-query
sc-status
sc-bytes
cs-version
cs(User-Agent)
cs(Referer)

Paso 0-2

Copie los contenidos de la carpeta cgi-bin de su disco su duro al directorio del cgi-bin de su servidor (esto incluye awststs.pl, awstats.model.conf, y los lang, lib y sud-directorios del plugins).

Paso 0-3

Pase subdirectories de icono de awstats y su contenido a un directorio leíble por su servidor por ejemplo C:\youwwwroot\icon.

Paso 0-4

Cree un archivo del config copiando awstats.model.conf archive en un nuevo archivo nombrado awstats.my virtualhostname.conf. Usted puede usar el valor de su opción en lugar del "myvirtualhostname". Este nuevo archivo debe guardarse en el mismo directorio que el cgi-bin de awstats.pl

Paso 0-5

Revise que estas nuevas configuraciones se archivan con su propio arreglo:

- Cambie el valor de LogFile con el path de los arhivos log de su servidor (Ud también puede usar el path de su directorio de awstats.pl).
- Cambie el valor de LogType con "w" para analizar los archivos log de su servidor, "S" para un archivo de log de servidor vertiendo "M" para los archivos de log de correo, "F" para el ftp anotan los archivos, "O" por otra parte.
- El logformat de cambio a un valor con el nombre de mismo campo definido en paso 0-1:

LogFormat = tiempo de la fecha c-ip los cs-username cs-método cs-uristem cs-uri-pregunta sc-estado sc-bytes el cs(User-agent)cs(Referer de la cs-versión) "

- El cambio el parámetro de Diricons para reflejar el path relativo de directorio del icono.
- Revise el parámetro de SiteDomain con el nombre del dominio principal o el de la intranet o el nombre del sitio web de su servidor (Example:ww.mydomain.com).
- El parámetro de AllowToUpdateStatsFromBrowser fijo a 1 si usted no tiene el acceso de línea de orden y tiene sólo acceso del cgi.
- Usted puede cambiar otros parámetros si usted quiere.

Actualizacion de Estadisticas

*** Paso 1-1:**

Deben hacerse la primera vez manualmente los primeros analyze/update de estadísticas de la línea del orden desde primer tiempo, el proceso, puede ser largo y es más fácil de resolver los problemas (si usted no tiene el acceso de Línea de Orden, simplemente va a Paso 1-2). Para poner al dia el awstas la orden es:

awstats.pl - el config=myvirtualhostname - la actualización

AWStats leerá los archivos config awstats.myvirtualhostname.conf (o si no encontró, awstats.conf) y crea la actualización de su base de datos con toda la información sumaria emitida del archivo del log analizado.

Se ahorran los archivos del AWStats de la base de datos en el directorio definido por el parámetro de DirData en el archivo del config.

Cuando la actualización está acabada, usted debe seguir la pantalla un resultado así:

Update for config "/etc/awstats/awstats.myvirtualhostname.conf"
With data in log file "/pathtoyourlog/yourlog.log"...
Phase 1 : First bypass old records, searching new record...
Searching new records from beginning of log file...
Phase 2 : Now process new records (Flush history on disk after 20000 hosts)...
Jumped lines in file: 0
Parsed lines in file: 225730
Found 122 dropped records,
Found 87 corrupted records,
Found 0 old records,
Found 225521 new qualifed records.

Los archivos dejados caer son archivos desechados porque ellos no eran ningún usuario que HTTP piden o las demandas no estaban calificadas por AWStats se filtra (Vea SkipHosts, SkipUserAgents, SkipFiles, OnlyHosts, OnlyUserAgents y parámetros de OnlyFiles). Si usted quiera ver qué líneas fueron dejadas caer, usted puede agregar el - la opción del showdropped en la línea del orden.

Los archivos corrompidos son archivos que no emparejan ningún formato del log definidos por el parámetro de "LogFormat" en el config/domain de AWStats el archivo. Con todo el webservers usted puede experimentar los archivos corrompidos un poco (Si todas sus líneas están corrompidas y LogFormat el parámetro en el AWStats config/domain archivo es correcto, puede ser el arreglo de formato de log en su servidor web que está equivocado. No haga olvídese que que su parámetro de LogFormat en el AWStats config/domain archivo debe emparejar el formato de archivo de log que usted analiza.

Si usted quiere ver que qué líneas están corrompidas, usted puede agregar el - la opción del showcorrupted en la línea del orden.

Los archivos viejos simplemente son archivos que ya se procesaron por un proceso de actualización anterior. Usted entendió que no es necesario para purgar su archivo del log después de cada proceso de actualización aun cuando es muy recomendado para hacerlo tan a menudo como sea posible.

Los Nuevos archivos son los archivos en su archivo del log que se usó con éxito a las estadísticas del build/update.

La nota: Un proceso de análisis de leño es lento (uno segundo para cada 4500 líneas de su logfile con Athlon 1Ghz, más la resolución de DNS,
tiempo para cada IP diferente se dirige en su logfile si DNSLookup se pone a 1 y no ya hecho en su archivo de log).

*** Paso 1-2:**

Pueden ponerse al día las estadísticas de AWStats de un navegador, proporcionar las estadísticas del real-tiempo, pulsando el botón la Actualización ahora" el eslabón que aparece cuando AWStats se usa como un CGI (la Próxima sección 'Lea las Estadísticas dinámicamente' le da URL para usar para que).

¡Advirtiendo!!

Para habilitar este eslabón, su parámetro AllowToUpdateStatsFromBrowser debe ponerse a 1 en su config archive (el Eslabón no se habilita por defecto).

Entonces, usando la actualización en línea no le impide frecuentemente ejecutar el proceso de actualización de un scheduler (el orden es mismo que ponga al día de primer proceso).

Para esto, usted tiene dos opciones:

- Incluya el orden de actualización en su proceso del logrotate. Vea FAQ-COM120 para esto.
- O agrega las instrucciones en su crontab (Unix/Linux) o su scheduler de la tarea (para Windows), para frecuentemente lanzar el Awstats
ponga al día el proceso.

Vea a AWStats Benchmark compaginar para la frecuencia de update/logrotate de recomendacion.

Leer las estadisticas

Ver los resultados de analizan, usted tiene varias soluciones que dependen de su política de seguridad.

* Primero la solución es construir los informes principales, en una página de HTML estática, de la línea del orden, así (el salto a segunda solución si usted tiene SÓLO CGI acceder):

el perl awstats.pl - el config=myvirtualhostname - el rendimiento - el staticlinks> awstats.myvirtualhostname.html

Usted puede usar todas las otras opciones del rendimiento (cada uno de ellos le dan otro informe). Esto es cómo usar todo el otro posible rendimiento el options(1):

perl awstats.pl -config=myvirtualhostname -output=alldomains -staticlinks > awstats.myvirtualhostname.alldomains.html
perl awstats.pl -config=myvirtualhostname -output=allhosts -staticlinks > awstats.myvirtualhostname.allhosts.html
perl awstats.pl -config=myvirtualhostname -output=lasthosts -staticlinks >

awstats.myvirtualhostname.lasthosts.html
perl awstats.pl -config=myvirtualhostname -output=unknownip -staticlinks >
awstats.myvirtualhostname.unknownip.html
perl awstats.pl -config=myvirtualhostname -output=alllogins -staticlinks >
awstats.myvirtualhostname.alllogins.html
perl awstats.pl -config=myvirtualhostname -output=lastlogins -staticlinks >
awstats.myvirtualhostname.lastlogins.html
perl awstats.pl -config=myvirtualhostname -output=allrobots -staticlinks >
awstats.myvirtualhostname.allrobots.html
perl awstats.pl -config=myvirtualhostname -output=lastrobots -staticlinks >
awstats.myvirtualhostname.lastrobots.html
perl awstats.pl -config=myvirtualhostname -output=urldetail -staticlinks >
awstats.myvirtualhostname.urldetail.html
perl awstats.pl -config=myvirtualhostname -output=urlentry -staticlinks >
awstats.myvirtualhostname.urlentry.html
perl awstats.pl -config=myvirtualhostname -output=urlexit -staticlinks >
awstats.myvirtualhostname.urlexit.html
perl awstats.pl -config=myvirtualhostname -output=browserdetail -staticlinks >
awstats.myvirtualhostname.browserdetail.html
perl awstats.pl -config=myvirtualhostname -output=osdetail -staticlinks >
awstats.myvirtualhostname.osdetail.html
perl awstats.pl -config=myvirtualhostname -output=unknownbrowser -staticlinks >
awstats.myvirtualhostname.unknownbrowser.html
perl awstats.pl -config=myvirtualhostname -output=unknownos -staticlinks >
awstats.myvirtualhostname.unknownos.html
perl awstats.pl -config=myvirtualhostname -output=refererse -staticlinks >
awstats.myvirtualhostname.refererse.html
perl awstats.pl -config=myvirtualhostname -output=refererpages -staticlinks >
awstats.myvirtualhostname.refererpages.html
perl awstats.pl -config=myvirtualhostname -output=keyphrases -staticlinks >
awstats.myvirtualhostname.keyphrases.html
perl awstats.pl -config=myvirtualhostname -output=keywords -staticlinks >
awstats.myvirtualhostname.keywords.html
perl awstats.pl -config=myvirtualhostname -output=errors404 -staticlinks >
awstats.myvirtualhostname.errors404.html

La nota (1): Si usted prefiere, usted puede usar los awstats_buildstaticpages labran con herramienta construir todas esas páginas en un orden o generar
PDF archiva.

La nota (2): Usted también puede agregar un filtro en el reports:url detail del rendimiento siguiente, el urlentry, el urlexit, el allhosts, el refererpages,el filtro puede ser un regexp en la llave llena usted quiere el awstats para presentar la información sobre y usted debe usarlo después del rendimiento parámetro separado por un ": ".

Por ejemplo, al rendimiento los urldetail informan, con un filtro en todas las páginas que contienen / las noticias, usted puede usar el orden siguiente
la línea:

el perl awstats.pl - el config=myvirtualhostname - el output=urldetail:/news - el staticlinks>
awstats.myvirtualhostname.urldetailwithfilter.html

La nota (3): Si usted quiere construir un informe durante un mes particular, agregue las opciones - el month=MM - el year=YYYY.

Para construir un informe durante el año lleno (advirtiendo: Esto puede usar mucha memoria y CPU), agregue las opciones - el month=all - el year=YYYY.

* Segunda solución es ver sus estadísticas dinámicamente de un navegador. Para esto, use URL:

http://www.myserver.mydomain/awstats/awstats.pl?config=myvirtualhostname

donde el myvirtualhostname se usa para saber qué config archivan para usar (AWStats usará awstats.myvirtualhostname.conf archivan).

La nota (1): Todas las rendimiento orden línea opciones (excepto - el staticlinks) todavía está disponible al usar AWStats como un navegador.

Simplemente úselos como los parámetros de URL como este ejemplo

http://www.myserver.mydomain/awstats/awstats.pl?month=MM=unknownos

La nota (2): Si el parámetro de AllowToUpdateStatsFromBrowser se pone a 1 en el config/domain de AWStats archive, usted también podrá correr el proceso de actualización de su navegador. Simplemente haga clic ahora" en la Actualización del eslabón.