Universidad Autónoma de Querétaro

Tesina

ORACLE HTTP SERVER

Alumno

Eduardo García Lara

Asesor

I.S.C. Jabel Resendiz González

Generación

1998-2002

CARTA DE ACEPTACIÓN

Por este medio, se otorga constancia de aceptación de tesina para obtener el título de Licenciado en Informática, que presenta el pasante **EDUARDO GARCÍA LARA**, con el tema denominado **"ORACLE HTTP SERVER".**

Este trabajo fué desarrollado como una investigación derivada del curso de titulación **"MODELADO DE BASE DE DATOS",** dando cumplimiento a uno de los requisitos contemplados en el artículo 34 del reglamento de titulación vigente, en lo referente a la opción de titulación por realización y aprobación de cursos de actualización.

Se extiende la presente para los fines legales a que haya lugar y para su inclusión en todos los ejemplares impresos de la tesina, a los diecisiete días del mes de Marzo del 2005.

ATENTAMENTE

ISC. JABEL RESÉNDIZ GONZÁLEZ
INSTRUCTOR DEL CURSO

Índice

Introducción.....3
1. Que es el servidor HTTP de Oracle y como trabaja.....7
1.1 Problemas de seguridad en el HTTP de Oracle.....7
1.2 Revelación del código de fuentes de los Scripts JSP... 3
1.3 Scripts DEMO.....8
1.4 Interfaces de Administración.....8
1.5 Módulo mod_plsql.....9
1.6 Uso de los paquetes predeterminados de PL/SQL... 9
1.7 Se realizan preguntas no autorizadas a la base..... 10
2. Cronología de los HTTP de Oracle.....10
2.1 Web Server 1.0 Oracle.....10
2.2 Web server 2.0 Oracle.....11
3. Servidor http de Oracle.....13
3.1 Oracle 9i Application Server Release 1.....14
3.2 El esquema cliente servidor en oracle.....15
3.2.1 Servicio.....15
3.2.2 Recursos compartidos.....15
3.2.3 Protocolos asimétricos.....15
3.2.4 Localización transparente.....15
3.2.5 Mezclados.....16
3.2.6 Basados en el intercambio de mensajes.....16
3.2.7 Servicios encapsulados.....16
3.2.8 Escalabilidad.....16
3.2.9 Integridad.....16
4. Aplicaciones Web.....17
4.1 Instrucciones para arrancar el servidor del HTTP de Oracle.....18
4.2 Cómo ejecutar programas de CGI del servidor de HTTP de Oracle.....19
4.3 Cómo una contraseña protege ciertos directorios.....19
4.4 Servidor de transacciones.....20
4.1.1 El puerto 80.....20
4.4.2 Oracle amplia el Web de Apache.....21
5. Configuración del servidor del HTTP de Oracle.....21
5.1 Archivos de la configuración del servidor del http.....22
5.2 Pasos para la configuración del servidor http.....23
5.3 Modificación de la configuración del servidor del http de Oracle por default.....26
5.4 Prueba de la configuración modificada del servidor del http.....28
6. Servicios de Comunicación.....29
6.1 Configuración del modulo del mod_plsql de Oracle.....29
6.2 Configuración del modulo del mod_cgi de Oracle.....29
6.3 Configuración del modulo del mod_perl de Oracle..... 30

7. Autentificación, autorización, y control de acceso en el servidor http.....31
7.1 Cómo trabaja la autentificación básica.....32
7.1.1 Crear un archivo de la contraseña.....33
7.1.2 Configuración de Apache para permitir el cgi.....37
8. Los servicios WEB.....38
8.1 Ventajas de los servicios WEB.....38
8.2 Razones para crear servicios WEB.....38
8.3 Inconvenientes de los servicios WEB.....38
8.4 Scripts Alias.....39
8.5 CGI fuera de los directorios Script Alias.....40
8.6 Archivos htacces.....41
9. Como escribir un programa del cgi para http.....41
9.1 Errores de Sintaxis.....42
9.2 Registros de errores.....42
9.3 Compilación de Apache.....43
9.4 Configurando el sistema de tiempo correctamente.....45
9.5 Programas de soporte para la compilación.....46
9.6 Errores durante el Start UP.....47
9.7 El comenzar en el cargador del tiempo.....47
9.8 Información Adicional.....48
10. Usar Apache Con Microsoft Windows.....48
10.1 Requisitos.....48
10.2 Descargar Apache para Windows.....49
10.3 Instalación de Apache pata Windows.....50
10.4 Archivos principales de la Configuración.....51
10.5 Los módulos.....54
10.6Alcance de los directorios.....55
11. La negociación en Apache.....57
11.1 Usar un archivo de tipo mapa.....57
11.2 Multiviews.....59
11.3 Los métodos de la negociación.....60
11.3.1 Existen dos métodos de la negociación.....60
11.4 Algoritmo de la negociación de Apache.....61
Conclusiones.....63
Bibliografía.....64
Indice de Imágenes.
Figura 1. Web Server 1.0 11
Figura 2. Web Server 2.0 12
Figura 3. Servidor de Aplicaciones.....17
Figura 4. Servicios que ofrece oracle 9i.....22

Introducción.

En los últimos años una forma muy popular de compartir la información son los servicios con las bases de datos. En tal arquitectura aparte del servidor Web se inician ciertos métodos dinámicos (p.ej. PHP, ASP, JSP) que descargan datos de las bases de datos y van generando las páginas Web. Todo crea una aplicación de Web. El cliente de esta aplicación es una navegadora de Web. Oracle y otros productores de las bases de datos han notado esta tendencia y han completado sus productos con la posibilidad de acceso a los datos mediante las navegadoras de Web. En el caso de Oracle el servidor Web es el Oracle HTTP Listener. Es la segunda (además del Listener predeterminado) forma de comunicarse con el servidor Oracle y, por lo tanto, también un lugar potencial de los ataques de fuera del sistema.

Oracle HTTP Listener es un muy bien conocido servidor Apache de la serie 1.3.x con los módulos añadidos por Oracle que lo integran con Oracle DBMS. El código ejecutado además del servidor se puede crear mediante varios métodos. Los más importantes son: PL/SQL – idioma de los procedimientos de Oracle; de su interpretación en caso de los scripts server-side es responsable el módulo mod_plsql.

Los scripts JSP y servlets Java – soportados por JServ (mod_jserv vinculado estáticamente con Apache) u Oracle Containers for Java (OC4J). Se puede decir que, en cuanto a los servidores de aplicaciones la Oracle opta por Java. Por lo tanto, los métodos son mucho más extendidos y de varias e interesantes posibilidades, p.ej. XSQL – integración con XML o bien SQLJ – ejecución directa de las preguntas SQL desde el interior de los scripts Java.

Otros métodos tradicionales, como, los scripts CGI ó Perl (mod_cgi y mod_perl están activos por default).

Hablaremos también de la importancia que tiene la arquitectura cliente/servidor y como interactúa con la tecnología de oracle ya que el movimiento cliente/servidor en computación es relativamente joven, Internet ha sido la encargada de darle un viraje total a este género de esquema. Hace pocos años los mainframes contenían sólo terminales brutas, en la actualidad y a través de estructuras como http este esquema esta revaluado, ya que para los servidores mainframes el trato con las PCs no es de este tipo, ellos necesitan

soportar protocolos punto a punto, interpretar los mensajes de la PC, ofrecer servicios a dichos PCs.

Para oracle y específicamente hablando del protocolo http es de suma importancia saber interpretar lo que es la arquitectura cliente/servidor ya que las aplicaciones cliente/servidor en la actualidad requieren un esquema híbrido de tal manera que puedan procesarse transacciones, diseño de bases de datos, experiencia de comunicaciones, uso de una Interfase gráfica de usuario GUI y uso de Internet. Las aplicaciones más avanzadas requieren conocimiento de objetos distribuidos e infraestructura de componentes. Los programadores necesitan tener este modelo en cuenta para producir software que se adapte a estas nuevas condiciones.

Un hecho que está redefiniendo el esquema cliente/servidor es el World Wide Web, ya no existe sólo la LAN entre dependencias de una empresa, sino conexión a WAN, esto significa que se necesitan esquemas de firewall, incluso, nuevos protocolos que soporten ecommerce. Lo que se conoce ahora como Extranet son un conjunto de túneles entre dos compañías sobre la Infraestructura pública de Internet para crear estos túneles se hace uso de Virtual Private Networks (VPN), la buena noticia es que en la actualidad existen estándares de Internet para crear estas VPN, y nuestro modelo cliente/servidor debe funcionar allí.

Oracle, IBM, Cisco, Hewlett Packard y Amazon entre otros han confiado en Apache debido a sus características de multiplataforma, estabilidad, escalabilidad, seguridad, performance y extensibilidad basada en la capacidad de agregar funcionalidad a través de módulos. Con respecto a esto último cabe destacar que PHP junto con ser uno de los lenguajes de programación para desarrollo web más utilizados del mundo es también el módulo más popular del servidor Apache.

Los proyectos del grupo Apache para Oracle , como el del servidor HTTP, giran en torno a Java, Perl y XML entre otros. The Jakarta Project (http://jakarta.apache.org) desarrolla y mantiene soluciones en plataforma Java capaces de satisfacer las más exigentes necesidades de entornos de explotación de tipo enterprise. Servidores de aplicación, frameworks, engines, herramientas, librerías y APIs componen la artillería Apache/Java donde destaca Tomcat, la implementación de referencia oficial para servlets y java server pages. El XML

Apache ha sido una solución ideal para los problemas orientados al servidor de http de oracle, ofrece la más amplia variedad de herramientas y de la más alta calidad.

Además, la Oracle comparte muchas tecnologías adicionales tales como buffers avanzados de las páginas dinámicas (Oracle Web Cache), framework para crear páginas de portales (Oracle Portal), implementaciones WebDAV y otros. El proceso Listener de http es s un programa que captura todas las solicitudes WEB las cuales son procedentes de un Browser cliente. Ya recibida la solicitud se determina el tipo de solicitud, esta puede ser una solicitud de Java, PL/SQL, HTML, entre otros, enseguida se reenvia esta solicitud al modulo apropiado del proceso Listener.

La primera aplicación cliente servidor que cubre todo el planeta es el World Wide Web. Este nuevo modelo consiste de clientes simples que hablan con servidores Web. Un servidor Web retorna documentos cuando el cliente pregunta por el nombre de los mismos. Los clientes y los servidores se comunican usando un protocolo basado en RPC, llamado HTTP. Este protocolo define un conjunto simple de comandos, los parámetros son pasados como cadenas y no provee tipos de datos. La Web y los objetos distribuidos están comenzando a crear un conjunto muy interactivo de computación cliente/servidor. Esta nueva convergencia se denomina los Objetos Web. Java applets y browser con soporte CORBA son las primeras manifestaciones de este tipo de Objetos Web. El próximo paso es la existencia de componentes Web. En la actualidad se da con los JavaBeans en el cliente y los Enterprise JavaBeans en el servidor. Estos clientes y servidores hablan unos con otros vía CORBA ó HHTP. En Microsoft, los objetos Web están basados en componentes ActiveX que se comunican vía COM+ o HTTP. Los servidores de aplicaciones Web son otra clase de software en Internet.

Es importante conocer lo que es un servicio Web lo cual interpretamos como una colección de protocolos y estándares que sirve para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferente y ejecutada sobre cualquier plataforma pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad

entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares

1. Qué es el servidor del HTTP de Oracle y cómo trabaja

El servidor del HTTP de Oracle es un servidor simple del httpd del Web. Se basa en el web server de Apache proporcionado por el grupo de Apache.
Las propiedades del servidor y de Oracle 9iAS (servidor de la base de datos de Oracle) con el servidor del HTTP de Oracle.

1.1 Problemas de Seguridad en el HTTP de Oracle.

No es difícil suponer que la mayoría de estos módulos adicionales poseen un largo y desarrollado listado de amenazas relacionadas con ellos. Es una situación típica para la aplicación que se desarrolla muy rápidamente. Está claro que en la instalación predeterminada está incluida la mayoría de estas extensiones. Según dice la práctica hay pocas instalaciones masivas que tengan configuración diferente de la predeterminada y los administradores por falta de sus conocimientos de todos los componentes avanzados prefieren dejarlos para no arriesgar la desestabilización de trabajo del servidor. Lo comprueba un antiguo principio que la seguridad es inversamente proporcional a la funcionalidad compartida por la aplicación. A se presentan casos de ataques interesantes que usan huecos en los módulos Oracle HTTP Listener. Todas las amenazas descritas se eliminan si se realiza una instalación correcta y supervisada.

1.2 Revelación del código de fuentes de los scripts JSP

Java Server Pages (JSP) es uno de los métodos mediante el cual el servidor puede generar las páginas Web dinámicas que representan la información cargada de la base de datos. Cuando el cliente (Browser) pida la página con la extensión .jsp, el servidor de forma predeterminada irá generando el código respectivo Java, lo irá compilando y ejecutando. El resultado en forma de la página HTML se devuelve a la navegadora. Durante el proceso en la pista /_pages del servidor Web se crean los archivos temporales. Uno de estos archivos con extensión java contiene el código de fuentes del scripto ejecutado.

En la configuración predeterminada del Apache distribuido con Oracle la carpeta /_pages es compartida por el servidor y, por lo tanto, el intruso puede leer el código de fuentes de las páginas JSP soportadas por el servidor.

1.3 Scripts demo

Muchas amenazas están relacionadas con los localizados en la instalación predeterminada de Oracle scripts que demuestran la funcionalidad de Oracle como servidor de aplicaciones. Estos scripts están localizados en la pista /demo. Muchos de estos ejemplos no son desgraciadamente modelos a seguir, en especial si se trata de los principios de la programación segura. Hay muchos scripts que demuestran la descarga de los datos desde la base a base de las variables procedentes del usuario que es susceptible a los ataques de tipo SQL injection. Una buena práctica de administrar obliga a eliminar de los servidores masivos cualquier funcionalidad y contenido innecesario, sin embargo, dejar los scripts demo es muy frecuente en las instalaciones masivas y muy populares en los servidores de desarrollo puestos en la Internet.

1.4 Interfaces de administración

Relativamente simples pero muy eficaces ataques se pueden realizar mediante las interfaces de administrar por las páginas Web. En la instalación predeterminada Oracle una gran parte de las páginas administradoras no requieren autenticación (sic!). Las URL-s de unas de estas páginas son las siguientes:

http://10.1.1.100/pls/admin_/gateway.htm,

http://10.1.1.100/oprocmgr-status,

http://10.1.1.100/servlet/Spy.

http://10.1.1.100/servlet/Spy.

1.5 Módulo mod_plsql

El módulo Apache mod_plsql sirve para interpretar en el servidor Web el código PL/SQL que es idioma nativo de las bases Oracle. En el módulo se han revelado muchos errores clásicos. Uno de ellos es rellenar el buffer entrante en el script que sirve para presentar ayuda. Enviar comando de tipo:

http://10.1.1.100/pls/simpledad/admin_/help/AAAAAAAA....(>1000 signos) este provoca error de segmentación del proceso que soporta este comando. Mediante la técnica buffer-overflow es posible ejecutar el código del intruso en el servidor, en el contexto del proceso del servidor Web.

Otro ataque al que es susceptible el mod_plsql, es aplicar la técnica double decode. Esta técnica está en enviar al servidor el comando que contiene caracteres especiales (como backslash) doblemente encriptados de forma hexadecimal. En consecuencia, es posible omitir las restricciones del servidor y leer cualquier archivo o carpeta dentro del servidor Web.Como por ejemplo, si queremos leer el archivo de configuración plsql.conf:

http://10.1.1.100/pls/simpledad/admin_/help/..%255Cplsql.conf

1.6 Uso de los paquetes predeterminados PL/SQL

La instalación predeterminada de Oracle la forma una grande librería que contiene varios paquetes de los procedimientos PL/SQL. En las versiones más antiguas de Oracle, todos los paquetes son accesibles también en la Internet mediante el mod_plsql. La llamada del procedimiento PL/SQL por el servidor Web es la siguiente:

http://ip.ip.ip.ip/pls/DAD/nombre del paquete.nombre del procedimiento

DAD (Database Access Descriptor) es estructura que describe la forma de conectarse con la base. En la instalación predeterminada es accesible un descriptor ejemplar llamado simplicidad.

El siguiente es un caso del ataque mediante los procedimientos del paquete owa_util.

Se Comprueba el paquete owa_util:

http://10.1.1.100/pls/simpledad/owa_util.signature

1.7 Se realizan preguntas no autorizadas a la base:

Otros paquetes PL/SQL que pueden ser interesantes son:

HTTP son procedimientos que permiten soportar el protocolo http.

TCP procedimientos que soportan el protocolo TCP, permiten entre otros abrir conexión reversible saliente.

FILE UNTIL son procedimientos de acceso a los archivos que permiten descargar cualquier archivo del servidor.

El administrador puede defenderse de usar estos paquetes al fijar el parámetro exclusion_list en el archivo de configuración wdbsrv.app. En la configuración predeterminada Oracle 9i este parámetro está fijado.

2. Cronología de los Web Servers http de Oracle

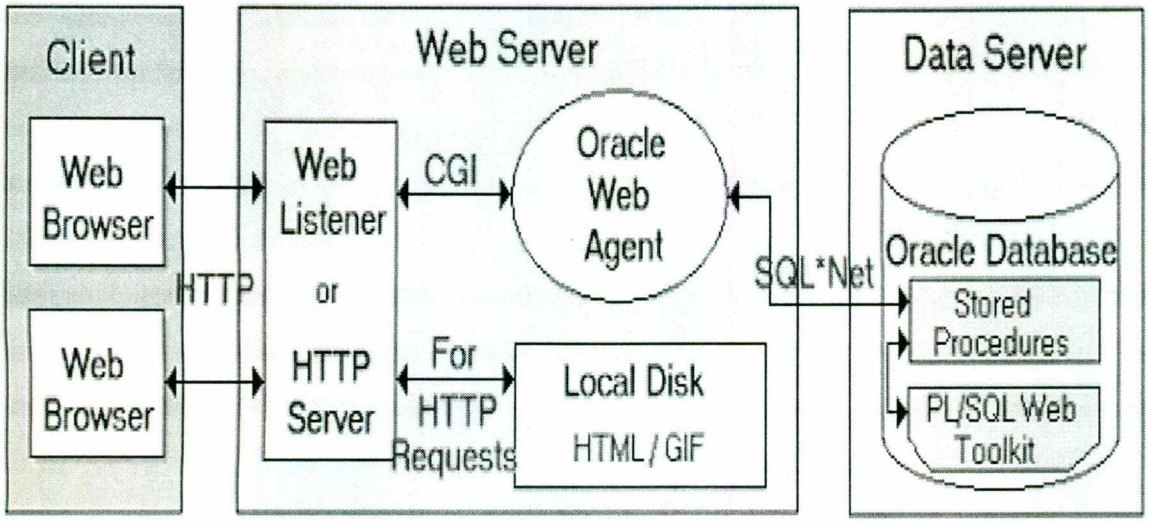
2.1 Web server 1.0 Oracle

El World Wide Web comenzó a ganar la aceptación en muchos negocios alrededor del mundo en 1995. Oracle respondió a este crecimiento trayendo el web server 1,0 de Oracle en el mercado.

El web server 1,0 de oracle no era nada más que un servidor del http (figura 1), a menudo llamado el oyente de la red, prestador de servicios de terceros. Como muchos otros servidores de la red durante este tiempo, el oyente de la red de Oracle proporcionó la ayuda para dos de los protocolos más comunes: Transporte del hypertext (HTTP) e interfaz de entrada común (cgi).

Fue incluido con el web server 1,0 de Oracle un programa del cgi nombrado el agente de red de Oracle (OWA). El oyente de la red invocaría el uso del cgi del agente de la red de oracle para ejecutar los procedimientos almacenados de PL/SQL que serían responsables de generar una respuesta del HTML para una petición del HTTP. Todos los procedimientos almacenados de PL/SQL utilizarían una caja de herramientas de la red de PL/SQL, también incluida con el web server 1,0 de Oracle, para generar una respuesta del HTML dinámicamente que contiene los datos recuperados de la base de datos, asegurando el documento entregado al web browser que contuvo datos actuales y actualizados.

La caja de herramientas de la red de PL/SQL es una colección de paquetes almacenados de PL/SQL usados para producir las etiquetas del HTML que contienen datos de la base de datos.



Web Server 1.0

Figura 1. Servidor HTTP

Uno de los obstáculos principales para este acercamiento era la escalabilidad. Para cada petición del HTTP, el oyente de la red creó un nuevo proceso de uso del cgi y presentó un coste enorme de frezar un proceso para cada petición. Esto era especialmente verdad para números más grandes de los pedidos simultáneos del cliente sistemas de la empresa.

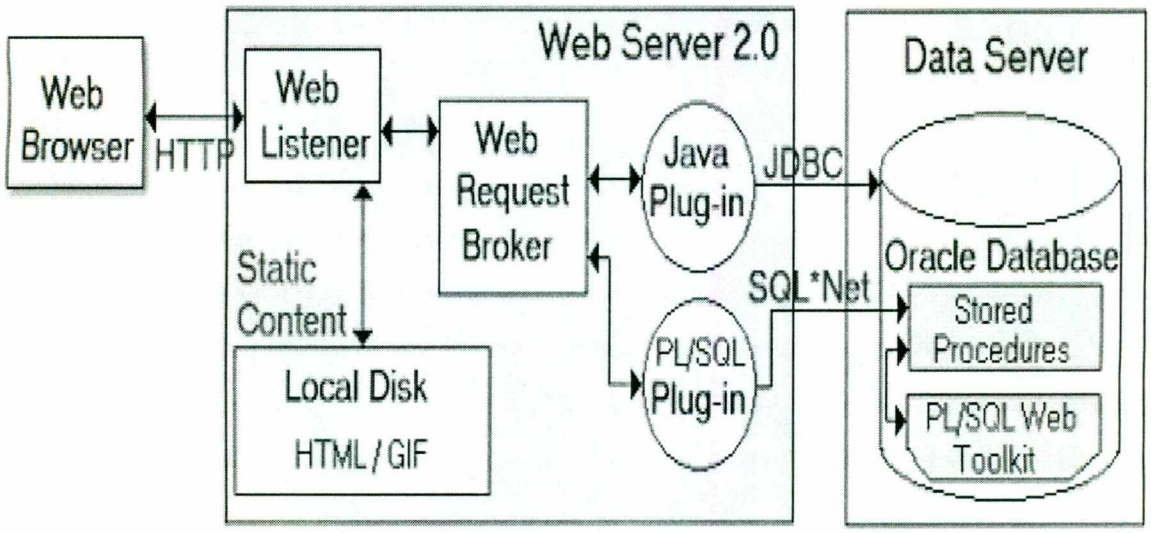
2.2 Web server 2,0 Oracle

Sabiendo que necesitaron tratar las ediciones inherentes en el cgi, de la escalabilidad el web server lanzado. Oracle reajustó totalmente su arquitectura y agregó la ayuda para los servicios enchufables.

Uno de los cambios más grandes con el web server 2,0 de Oracle era la inclusión de un proceso persistente llamado el corredor de la petición de la red (figura 2). Un interfaz de programación del uso (API) fue proporcionado el corredor de la petición de la red que fue utilizado para construir usos enchufables. El WRB entonces actuaría como interruptor para dirigir todas las peticiones del HTTP a un uso enchufable específico. El uso del cgi de

OWA, incluido con el web server 1,0 de Oracle que ejecutó procedimientos almacenados de PL/SQL fue convertido en un plug-in llamado el agente de PL/SQL.

También fue incluido con el web server 2,0 de oracle *un plug-in* de Java para ejecutar los usos de Java para dinámicamente generar el HTML. El plug-in de Java fue basado en JDK 1.1.2 e incluyó una máquina virtual integrada de Java (JVM). El JVM incluido con OWS 2.0 tenía la entrada estándar y las corrientes de la salida rewired para comunicarse con el WRB. En cortocircuito, el corredor de la petición de la red dirigiría todas las peticiones del HTTP al plug-in del agente o de Java de PL/SQL. El WRB entonces vuelve todas las respuestas del HTTP de nuevo a la opinión del cliente a través del oyente de la red.



Web Server 2.0

Figura 2. Se añade el corredor de la petición de la red

Otro plug-in incluido para este servidor incluye (SSI) era el plug-in de livehtml Oracle también publicó el corredor API de la petición de la red para permitir que cualquier persona construya un uso enchufable. Estos enchufables son lo que el web server 2.0 de oracle a hecho en *un ambiente* extensible. Un proceso del corredor de la petición de la red (WRB) fue asignado a cada oyente de la red y se ejecutaba ya persistente antes de que llegara una petición del HTTP. Los gastos indirectos de comenzar un nuevo proceso para cada petición

del HTTP fueron eliminados, al igual que el caso en la puesta en práctica del cgi de la orden en el web server 1.0 de Oracle y de tal modo de mejorar funcionamiento al procesar peticiones del HTTP.

Escribir un uso de Java para el web server 2.0 de oracle era muy simple y directo. Todo lo que el revelador necesitaría hacer debía proporcionar un método estático público estándar del main() en su clase de Java, y utiliza System.out.println() para escribir el HTML generado a la salida estándar.

Cuando el web server 2,0 de oracle fue lanzado, Servlet API era todavía el líder en el medio y el web Server de Oracle todavía no había sido publicado. El web server 2.0 de oracle era realmente el primer paso de oracle en el mundo de java. Oracle también aproximadamente este tiempo licenció la tecnología de JBuilder de Borland para proveer de los reveladores una herramienta de desarrollo de Jav. Oracle primero nombró la herramienta de desarrollo AppBuilder para Java pero más adelante la retituló JDeveloper .

3. Servidor http de Oracle

La búsqueda de Oracle para ampliar más lejos su escalabilidad en una tecnología del servidor del uso continuó con su lanzamiento del servidor 4.0 de oracle. Los procesos del cartucho de OWAS 3,0 ahora son procesos multiroscados llamados los servidores del cartucho, o procesos de uso. Es posible que un proceso del servidor del cartucho funcione casos múltiples del cartucho del mismo tipo como por ejemplo si un servidor del cartucho de PL/SQL puede funcionar cartuchos múltiples de PL/SQL con cada cartucho que mantiene una petición del HTTP. Esto era una mejora importante OWAS excesivo 3.0 donde estaba un proceso cada cartucho que manejaba una petición del HTTP separado. Otros cambios importantes en la arquitectura de OWAS 4.0 incluida

El encargado del nodo ahora proporcionó control de la administración sobre una instalación del multi-nodo OAS de una sola localización.

El cartucho de JServlet ahora apoyó el sol Servlet API y substituye el cartucho original de Java, el cartucho de JWeb.

Se agregó soporte al Apache Wbe listener en UNIX.

La maquina virtual de Java fue actualizada para soportar JDK 1.1.6.

3.1 Oracle 9i Application Server Release 1

El release 1 del servidor del Oracle9 es la nueva plataforma del servidor del Oracle y sigue encendido de OWAS 3.0 con mejor funcionamiento y escalabilidad que cualquier lanzamiento anterior. Oracle9iAS utiliza el software del servidor del Web de Apache para todo el tráfico del HTTP y el módulo de Apache JServ para los servlets de Java.
La herramienta de PL/SQL en Oracle9iAS escrito como módulo de Apache, llamado una MOD. Es decir mod_plsql que ahora se llama la entrada de PL/SQL

Además de los módulos de Apache, Oracle proporciona su propio motor de la página del servidor de Java JSP para el servidor HTTP, y el Oracle9i JVM para ejecutar la empresa JavaBeans o componentes de CORBA.

Para funcionar el Oracle9i se necesita solamente instalar una instancia de la base de datos de Oracle con la opción de software de Apache para ordenar peticiones a Oracle cuando está requiriendo. Una configuración detallada en este tipo de servidor, lo proporciona Java el cual despliega Servlets y a JSPs usando el servidor del HTTP de Oracle.

Como leer la configuración XSQL mediante el servlet XSQLServlet
El archivo XSQLConfig.xml contiene información sobre la configuración de las extensiones XSQL, entre otros el nombre y la contraseña del usuario con cuya autorización el módulo XSQL se conecta con la base de datos. Este archivo se encuentra en el espacio del servidor Web, en la pista /xsql/lib/XSQLConfig.xml. El servidor prohíbe acceso a este archivo y una prueba directa de acceso al archivo causa devolución de la información. Como este es el archivo XML se puede leerlo usando el servlet XSQLServlet compartido predeterminadamente:
http://10.1.1.100/servlet/oracle.xml.xsql.XSQLServlet/xsql/lib/XSQLConfig.xml

3.2 El esquema cliente servidor en oracle.

En realidad no hay un consenso sobre lo que el término significa, puede verse como entidades lógicamente diferentes que trabajan sobre una red para cumplir una tarea. Para el servicio que ofrece Oracle a través de servidor de http es sumadamente importante conocer los conceptos y la terminología de lo que es la estructura cliente/servidor y así saber lo que sucede atrás de esta estructura bajo este esquema. Al parecer todos los sistemas cliente/servidor deben tener las siguientes características que los distinguen de los demás esquemas:

3.2.1 Servicio

Un esquema cliente/servidor puede verse como una relación entre procesos corriendo o ejecutándose en máquinas separadas. El servidor es un proveedor de servicios. El cliente es un consumidor de servicios. En esencia, el esquema cliente/servidor provee una clara separación basándonos en la idea del servicio.

3.2.2 Recursos compartidos

Un servidor puede atender muchos clientes al mismo tiempo y regular el acceso de los mismos a los recursos compartidos.

3.2.3 Protocolos asimétricos

Existen relaciones muchos a uno entre los clientes y un servidor. Los clientes siempre inician el diálogo solicitando el requerimiento a un servicio. Los servidores están siempre pasivos esperando los requerimientos de los clientes.

3.2.4 Localización transparente

El servidor es un proceso que puede estar en la misma máquina que el cliente o en diferentes máquinas sobre la red. El software cliente/servidor frecuentemente enmascara la localización del servidor para los clientes, redireccionando las llamadas al servicio cuando se hace necesario. Un programa puede ser un cliente, un servidor o ambos.

3.2.5 Mezclados

El software ideal en el esquema cliente/servidor es independiente de la plataforma de hardware y del sistema operativo utilizado. Pueden mezclarse plataformas cliente/servidor.

3.2.6 Basados en el intercambio de mensajes

Un esquema cliente servidor debe estar acoplado e interactuar en un mecanismo de paso de mensajes. Los mensajes son usados para solicitar y recibir un servicio.

3.2.7 Servicios encapsulados

El servidor es un especialista. Un mensaje le dice al servidor que servicio es requerido, entonces el servidor determina como realizará su trabajo.

3.2.8 Escalabilidad

Los sistemas cliente/servidor deben escalarse tanto horizontal como verticalmente, horizontalmente significa poder adicionar o retirar estaciones de trabajo con muy bajo impacto sobre el rendimiento. Verticalmente, significa migrar a estaciones más grandes y rápidas o a sistemas distribuidos sobre la red.

3.2.9 Integridad

El código en el servidor y los datos en él están administrados de forma centralizada, lo cual ofrece integridad y seguridad a los datos. Al mismo tiempo, los clientes son personales e independientes.

Muchos sistemas con arquitecturas diferentes se han denominado cliente/servidor. Incluso vendedores de sistemas aplican el concepto solamente a sus paquetes específicos, se llama cliente servidor a servidores de archivos, servidores de bases de bases de datos, objetos distribuidos, monitores TP, esquemas groupware, a Internet y toda su lista de tecnologías, la pregunta es ¿Cuál de éstas tecnologías es realmente cliente/servidor? Varias formas de soluciones en red se han denominado de este tipo, cada una de estas soluciones se distingue por el tipo de servicio que ofrece. En realidad la esencia de cada uno de estos tipos se halla

en el middleware, más tarde se revisará este concepto, por ahora veamos los diferentes tipos de servidores.

4. Aplicaciones Web

Se acaban de mencionar algunos ejemplos de ataques a las aplicaciones que integran la base de datos Oracle con la Internet. A pesar de todo, los errores cometidos por el productor de la plataforma que es el servidor de la aplicación no son tan importantes como los errores cometidos por los creadores de las aplicaciones Web compartidas por este servidor. El corazón de cada instalación de tipo servidor de aplicaciones es una aplicación creada exactamente para las determinadas necesidades (figura 3). Estas aplicaciones en forma de scripts PL/SQL, JSP, servlets Java etc. operan en los servidores Web y se comunican con la base de datos con la ayuda del protocolo HTTP. Hay que recordar que cada error cometido por el programador en este tipo de scripts puede afectar con resultados incalculables. Los scripts son peligrosos ya que la esencia de su funcionalidad es interacción con muy a menudo anónimo usuario procedente del Internet. Como utilizan el protocolo HTTP, las amenazas relacionadas con ellos no se pueden controlar mediante los mecanismos tradicionales como los Firewall e IDS. La única forma de proteger es un detallado control del código por especialistas adecuados.

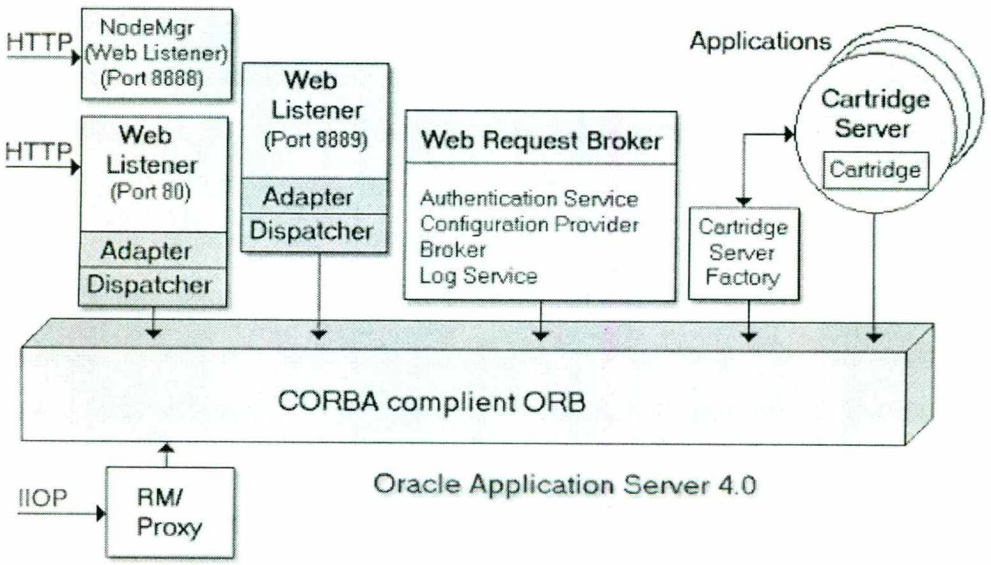


Figura 3.servidor de aplicaciones

Cabe mencionar que la configuración predeterminada de Oracle puede facilitar la tarea al intruso. Un ejemplo muy curioso es generalmente conocido cuentas y contraseñas de la instalación predeterminada de Oracle. En Internet podemos encontrar el listado de más de 100 cuentas instaladas predeterminadamente por Oracle 8i y 9i además de los productos que lo acompañan. En un principio son cuentas interiores del sistema o cuentas relacionadas con las aplicaciones demo. En el Oracle 9i unas de estas cuentas tienen autorizaciones muy altas:

CTXSYS (contraseña CTXSYS) – autorización DBA (administrador de la base),

TRACESVR (contraseña TRACE) – authorization select any table,

MDSYS (contraseña MDSYS) – set de autorizaciones muy altas.

Lo característico es que estas cuentas están relacionadas con la funcionalidad marginal de CTXSYS es cuenta necesaria para la actividad del paquete necesario a la búsqueda contextual de documentos y MDSYS para soportar los datos multidimensionales.

Oracle recomienda eliminar o bloquear las cuentas claves si su funcionalidad no se usa, aunque, un gran número de los administradores, no sólo de Oracle, no tienen en cuenta estas sugerencias o no las toman en cuenta.

4.1 Instrucciones para arrancar el servidor del HTTP de Oracle

Incluso se recomienda utilizar el comando apachectl que el comando del httpdsctl pues el apachectl también fija un número de variables de entorno requeridas. Cómo publicar las páginas estáticas del HTML en el servidor del HTTP de Oracle se puede utilizar el servidor de HTTP de oracle para publicar las páginas estándares del HTML. Este comienza definiendo un alias o también llamado directorio virtual en tela, al punto de directorio físico que contiene las páginas del HTML que se necesitan publicar.

Acceder a la siguiente ruta: cd $ORACLE_HOME/Apache/Apache/conf

Necesitamos corregir httpd.conf y agregar las líneas siguientes:

Alias/mydocs/"/mi/directorio/nombre/"
<Directory"/mi/directorio/nombre " >
Permitir de a ll6 </Directory>
Reiniciar el servidor del HTTP de Oracle.
Ahora Abrimos WEB browser y navegamos a:
http://my.host.name:7777/mydocs/index.html

4.2 Cómo ejecutar programas de CGI del servidor de HTTP de Oracle

Uno puede ejecutar cualquier programa que se conforme con el estándar del cgi (interfaz de entrada común) del servidor del HTTP de Oracle. Tal programa se puede escribir en el Perl, el TCL, C++, COBOL o cualquier otro lenguaje de programación.

Para lograr esta interacción del cgi con http debemos agregar la línea siguiente en el archivo de httpd.conf y reiniciar el servidor:

ScriptAlias/cgi-compartimiento/"/usr/local/apache/cgi-compartimiento/"

4.3 Cómo una contraseña protege ciertos directorios

Necesitamos agregar los directorios siguientes al Directory entrada en el archivo de httpd.conf y reiniciar el servidor de Apache. Se puede también agregar estos directorios al $ORACLE_HOME/a Apache/archivo del modplsql/del cfg/plsql.conf para proteger TODO EL acceso de base de datos:
AuthName "autentificación" AuthType AuthUserFile
Podemos utilizar la utilidad del htpasswd para crear el archivo de la contraseña.

htpassswd - c/la trayectoria//su/contraseña/el archivo username1 # crea el htpassswd/la trayectoria iniciales del archivo del passowrd//su/contraseña/el archivo username2 #, este agrega a segundo usuario al archivo del passowrd

Podemos observar que se pueden agregar mas usuarios sin necesidad de reiniciar el servidor de Apache. Existen algunas escrituras prácticas del Perl CGI-BIN están disponibles para manejar el registro del usuario de un Web page.

4.4 Servidor de transacciones

Con un servidor de transacciones, el cliente invoca procedimientos remotos (ó servicios) ellos residen en el servidor el cual contiene el motor de la base de datos SQL. Estos procedimientos remotos en el servidor ejecutan un grupo de tareas SQL. El intercambio en la red consiste de un mensaje request/reply (opuesto al anterior donde existe un mensaje request/reply por cada transacción SQL). Estos requerimientos SQL agrupados, se denominan transacciones. Con un servidor de transacciones, se debe crear la aplicación cliente/servidor para ambos componentes. El componente para el cliente usualmente incluye una interfase gráfica de usuario, GUI. El componente en el servidor consiste del servidor de transacciones de la base de datos. Este tipo de aplicaciones se denominan Online Transaction Processing ó OLPT. Las OLPT requieren alto control de seguridad e integridad de los datos, dos formas de OLPT son TP-Lite, basado en la carga de procedimientos provistos por los vendedores de bases de datos y TP-Heavy basado en monitores TP. TP = Transaction Processing Monitors, especializados en administrar transacciones desde el punto origen típicamente el cliente a través de uno o más servidores y retornar el resultado al cliente. Cuando una transacción finaliza los monitores TP aseguran que todos los sistemas envueltos en la transacción se encuentran en un estado consistente, ellos saben como ejecutar la transacción, la ruta de las transacciones a través de todo el sistema, el balance de carga para su ejecución y como restaurarla en caso de fallas.

4.4.1 El puerto 80.

Los puertos debajo de 1024 en los sistemas de Unix y de Linux son puertos privilegiados; ya que solamente las cuentas con permisos del súper usuario, como raíz, pueden comenzar programas que realmente sean de utilidad. Si es necesario que el servidor de Apache utilice

el puerto 80 por default, necesitamos que el administrador del sistema nos permita realizar las operaciones necesarias.

4.4.2 Oracle amplía el Web Server de Apache

El web server de Apache puede ser ampliado escribiendo los módulos o en el caso de Apache los mods. Oracle proporciona los siguiente mods:

Oracle PL/caja de herramientas del SQL (mod_plsql). Se requiere ejecutar el PL/el SQL en el DB y el HTML de la vuelta a los browsers.

Estos son algunos de los mods estándares de Apache que pueden ser utilizados:

mod_auth: autentificación básica del Web

mod_perl: Utiliza las escrituras del Perl más rápidamente que usan al intérprete del Perl de Apache

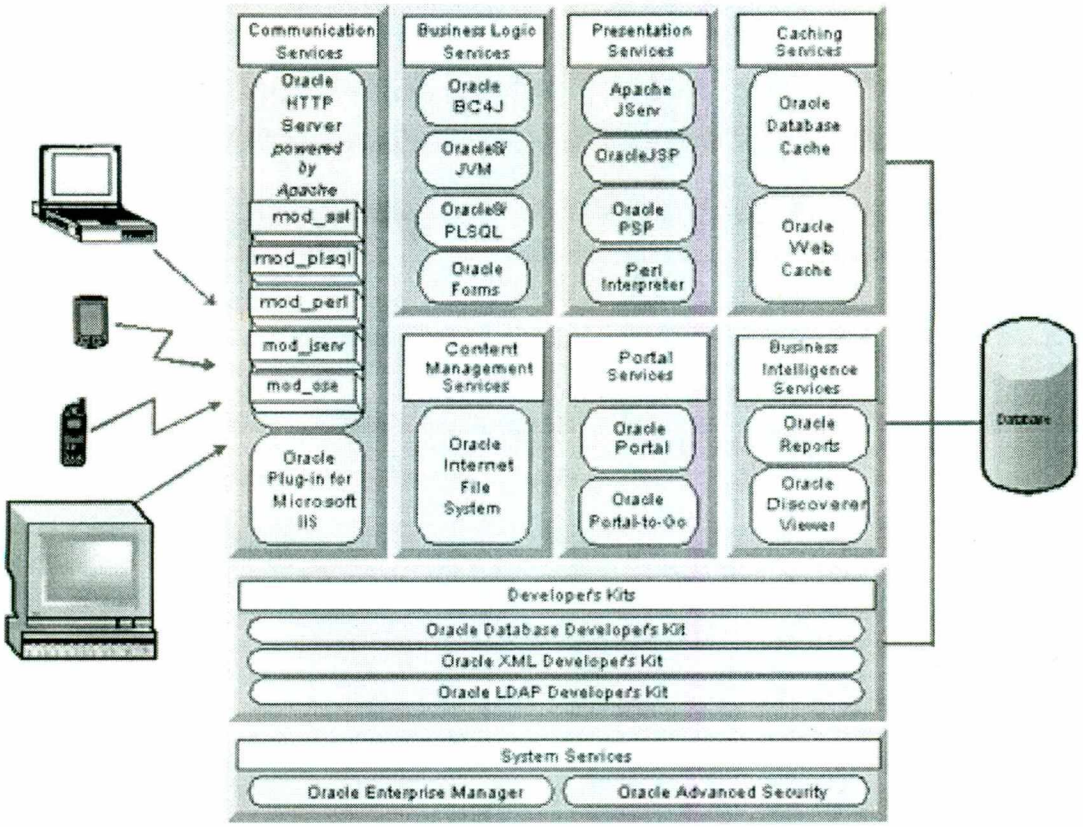
mod_ssl - asegure la capa del zócalo (el SSL)

mod_cgi - funcione las escrituras y los programas vía el interfaz de entrada común.

5. Configuración del servidor del HTTP de Oracle.

El servidor del HTTP de oracle es configurado corrigiendo el archivo de httpd.conf en su $ORACLE_HOME/Apache/Apache/directorio del conf a mano. Oracle no proporciona ninguna utilidad del GUI para corregir este archivo. Es necesario consultar la documentación estándar de Apache antes de realizar cualquier cambio a este archivo.

El servidor http de Oracle9i es confiable, escalable, seguro, este servidor se diseña para apoyar la evolución de una compañía o un negocio (figura 4). Proporciona un sistema de servicios para manejar la complejidad tecnológica de montar una infraestructura completa del Internet. Oracle9i proporciona una infraestructura que pueda crecer con las compañías.



Servicios que ofrece oracle9i

Figura 4. El servidor HTTP es confiable, escalable, seguro

5.1 Archivos de la configuración del servidor del HTTP

En Windows NT, los archivos de la configuración del servidor del HTTP de Oracle están en una localización por default en el ORACLE_HOME \ Apache \ Apache \ conf.

El archivo principal de la configuración se llama httpd.conf. Se debe utilizar solamente este archivo para configurar el servidor del HTTP, porque es mucho más fácil manejar el archivo de la configuración si hay solamente uno. No se debe utilizar srm.conf o access.conf.

5.2 Pasos para la configuración del servidor HTTP

Ejecución de las tareas de Pre configuración.

Bajo la plataforma NT solamente un servidor del HTTP de Oracle puede funcionar a la vez. Ya que se hacen servidores múltiples del HTTP al instalarlo, nos debemos cerciorar de que los otros servidores no estén funcionando. Sin embargo, se deben realizar los pasos siguientes:

1. Seleccionar los servicios del panel de control.

2. Buscar el servicio de OracleOraHome91HTTPServer. Detener y hacer doble clic sobre el (figura 5).

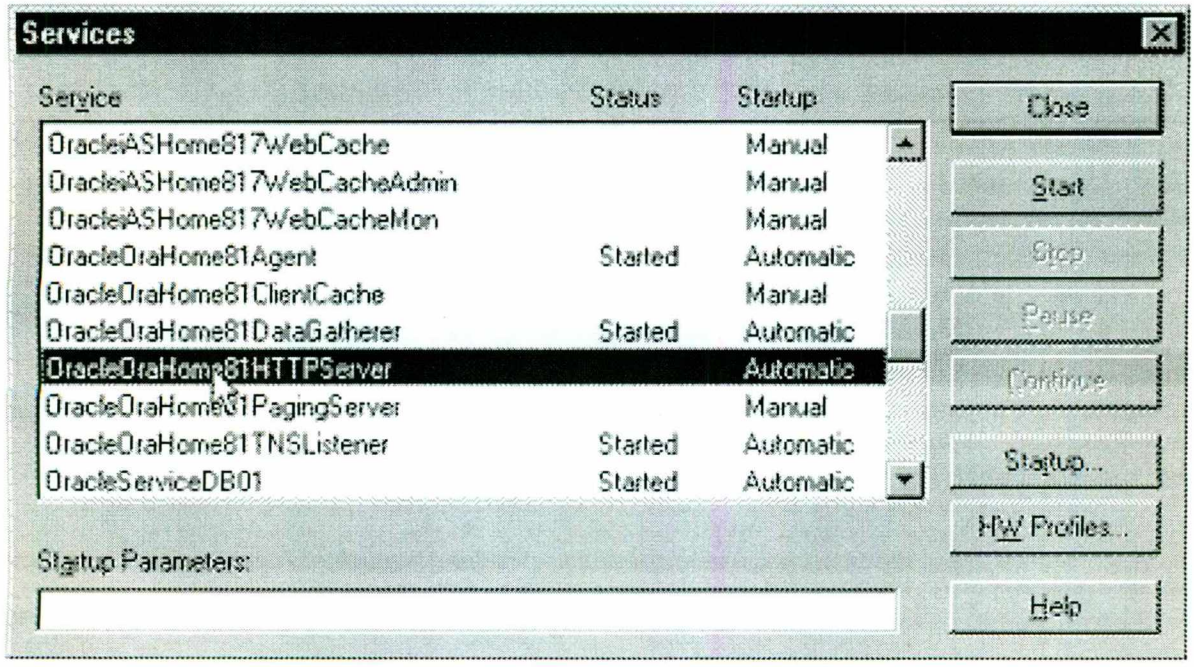


Figura 5. Buscando el servicio OracleOraHome91HTTPServer

3. Cambiar el tipo de arranque a manual (figura 6).

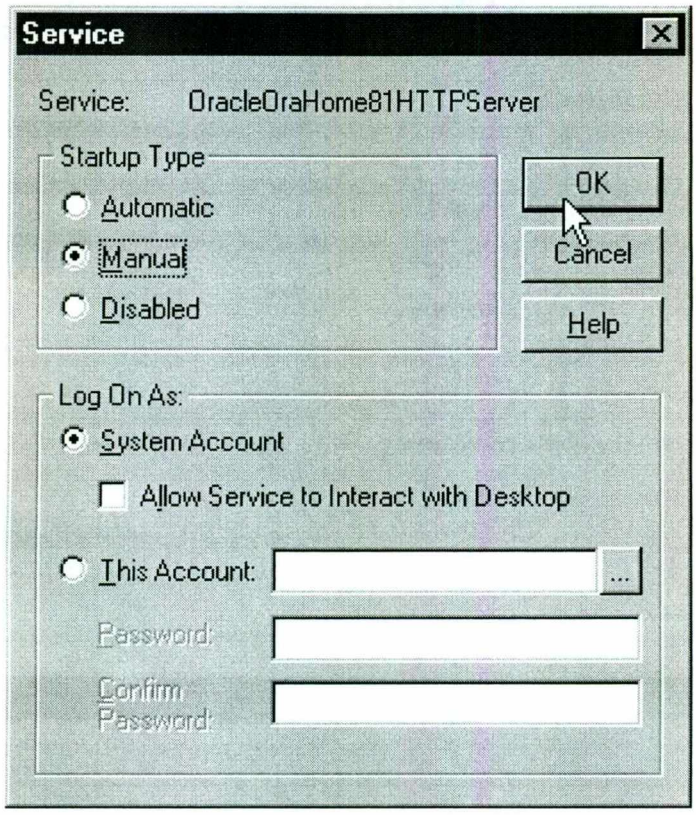


Figura 6. Cambiar el arranque a manual.

4. Cerciorarnos de que el único servicio del servidor del HTTP que se fija a automático sea OracleiASHome917HTTPServer (figura 7).

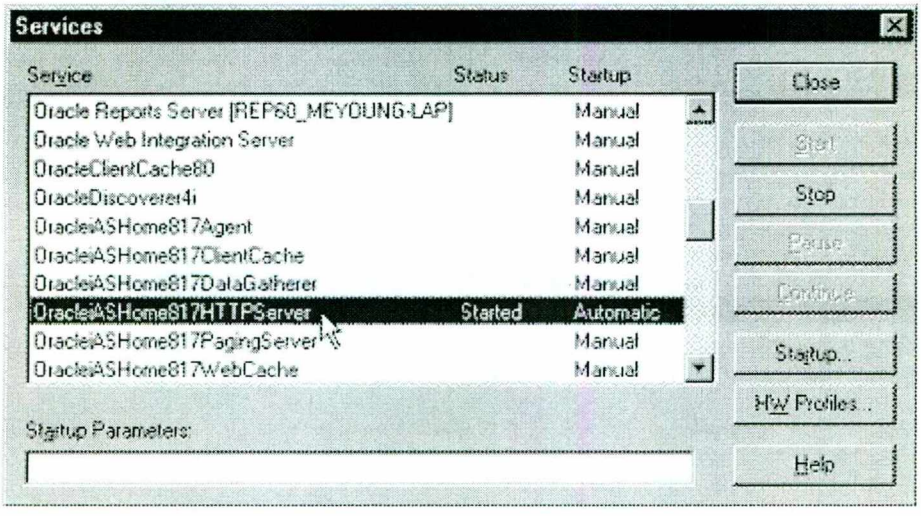


Figura 7.Revisar los servicios del servidor de HTTP

5. Cerciorarnos de que el archivo tnsnames.ora tenga un alias para señalar a la base de datos donde se instalaron los datos que son necesarios para Oracle9iAS (figura 8)

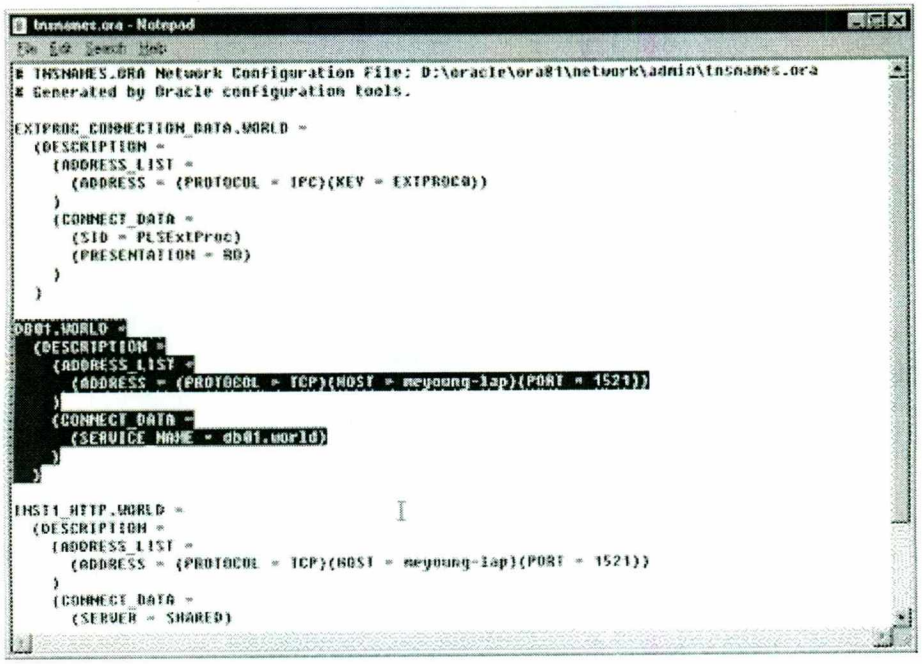


Figura 8. Revisar que el archivo tnsnames.ora tenga un alias tenga un alias

6. Abrir el archivo de tnsnames.ora de orácle iASHome817. Pegar la entrada de db01.world (figura 9).

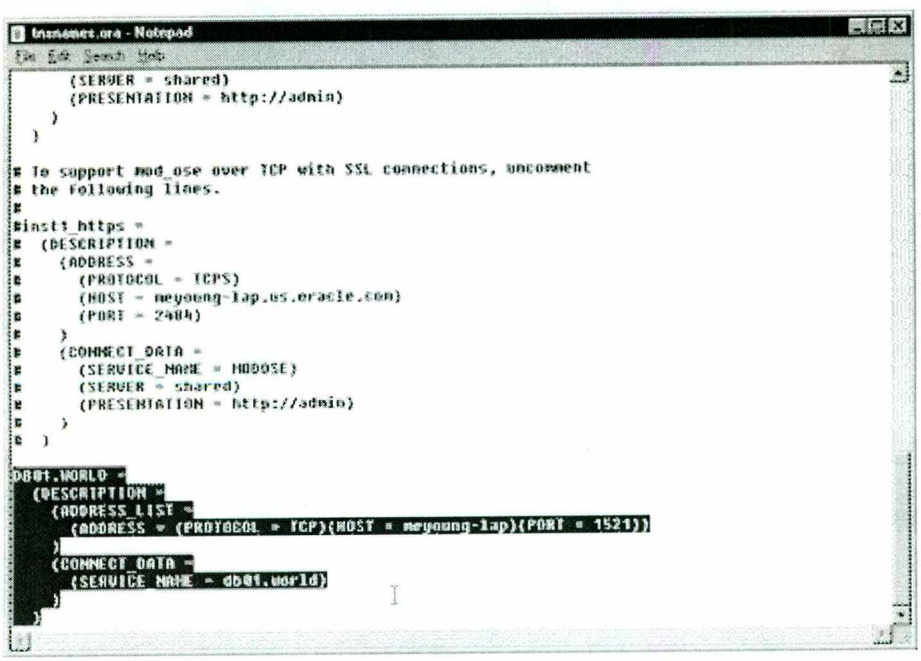


Figura 9. Abrir el archivo de tnsnames.ora

7. Reiniciar el sistema

Una vez que el sistema haya reanudado, se debe verificar el servidor de HTTP de Oracle (Figura 10). Esto se logra usando el browser, y entramos en http://hostname:7778 en la localización del URL.

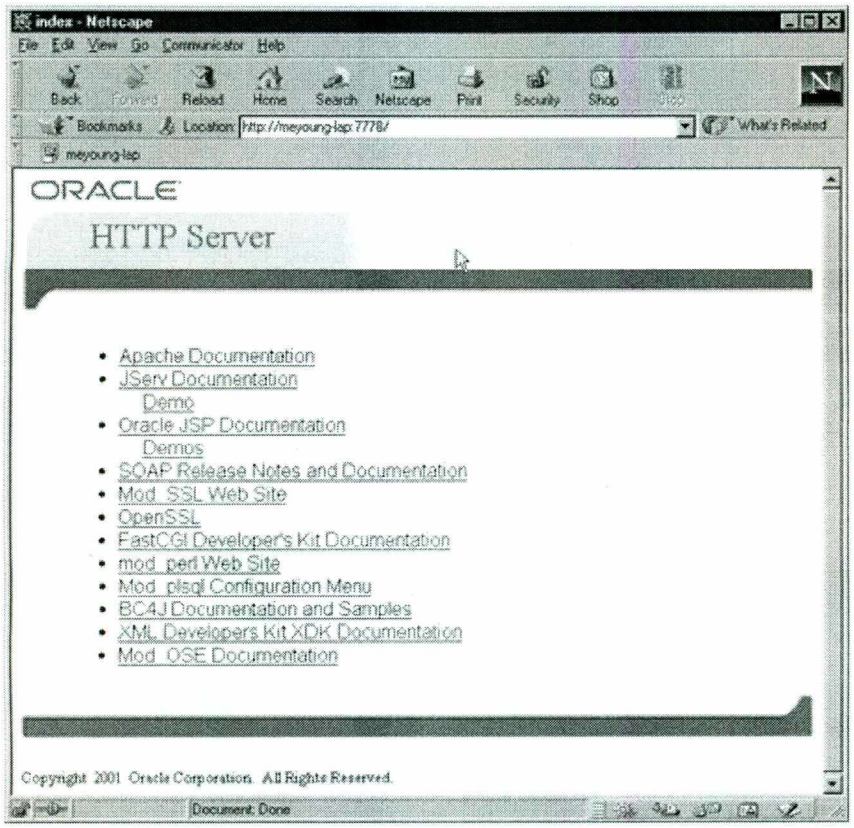


Figura 10. Verificando el servidor de HTTP.

5.3 Modificación de la configuración del servidor del HTTP de Oracle por default.

Después de la instalación, se debe corregir el archivo de httpd.conf y especificar el puerto por default del administrador, y la localización del contenido estático. Además, debemos agregar un alias a un directorio físico para permitir el acceso a un sistema específico de archivos.

1. Abrir % de ORACLE_HOME % \ Apache \ Apache \ conf \ httpd.conf en libreta.

2. Búsqueda para el PUERTO 7778 de la frase. Cambiar el número de acceso a partir del 7778 a 80.

3. La búsqueda para la frase ESCUCHA 7778. Cambiar el número del escuchar a partir del 7778 a 80.

4. Busque para ServerAdmin y cámbielo de you@your.address a la dirección del E-mail requerida.

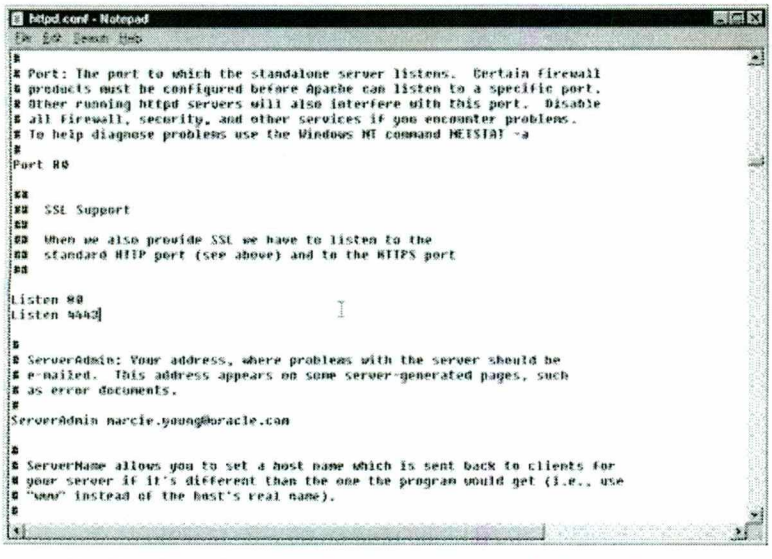


Figura 11. Cambiando la configuración por default del Server Admin

5. Búsqueda para alias (figura 12). Agregar un alias que señale al directorio donde se localizan los archivos del documento, incorporando esta declaración: Alias /mydocs/ D:\oracle\ora81\Apache\mydocs/"6. Guardar el archivo y salir.

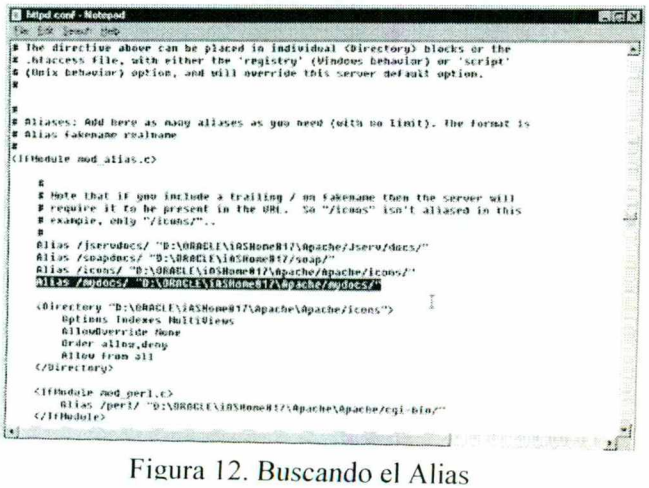


Figura 12. Buscando el Alias

5.4 Prueba de la configuración modificada del servidor del HTTP

1. Se necesita parar y correr el servidor del HTTP. Para hacer esto, debemos entrar a los servicios de http y detener el mismo. Después espere cerca de 30 segundos y arrancamos el servidor seleccionando.

2. Para probar el servidor del HTTP de Oracle. En el browser, incorporamos el hostname de http:///:80 en la localización del URL:

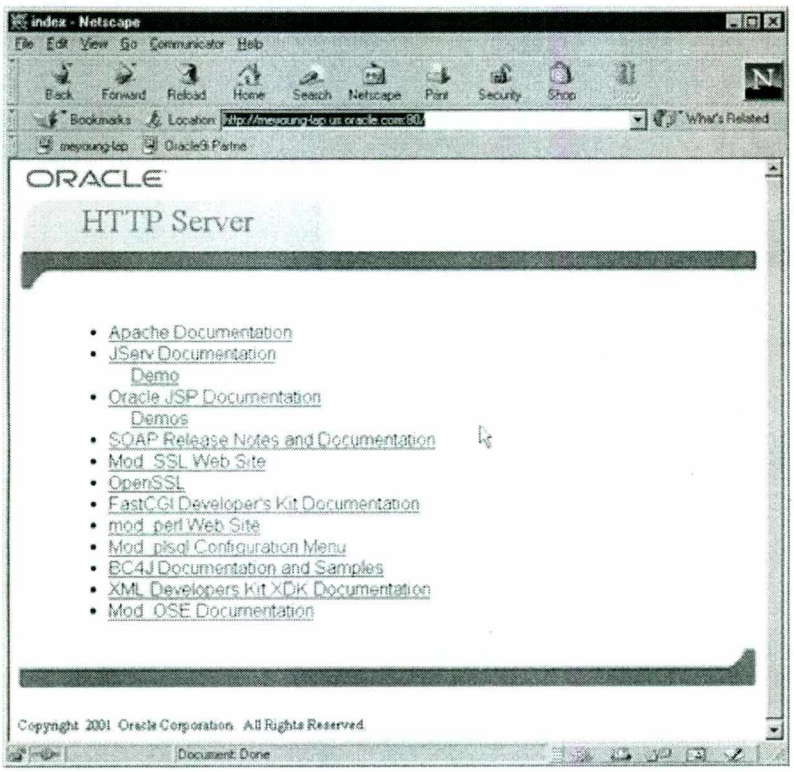


Figura 13. Probar el servidor del HTTP.

La misma ventana aparece como la que se verificó anteriormente; sin embargo, esta vez está conectada con el puerto 80.

6. Servicios de Comunicación.

6.1 Configuración del módulo del mod_plsql de Oracle.

Este módulo, mod_plsql, permite al servidor de Oracle9i conectar con un servidor de la base de datos de oracle y ejecutar procedimientos almacenados. Cada petición del mod_plsql se asocia a un descriptor del acceso de base de datos (PAPÁ), el cual es un sistema nombrado de valores de la configuración usados para el acceso de base de datos. UN PAPÁ especifica la información tal como la base de datos alias y una secuencia del conectar si la base de datos está alejada. Los procedimientos del PL/SQL, invocados de mod_plsql, pueden realizar algunas operaciones en la base de datos y volver los resultados al usuario, o genere las páginas dinámicas del HTML que contienen datos de la base de datos. Este módulo se recomienda para los procedimientos apátridas del PL/ SQL, donde el estado de la transacción y los valores de las variables del paquete no se preservan.

6.2 Configuración del módulo del mod_cgi de Oracle.

El directorio de ScriptAlias en el archivo de httpd.conf se utiliza para marcar el directorio que contiene las escrituras del CGI. Es realmente idéntico al directorio más genérico del alias. Un ejemplo de este directorio es el siguiente:

ScriptAlias/cgi-compartimiento/"D:\oracle\ora91\Apache\Apache\cgi - compartimiento/"

Se puede cambiar por default al punto a cualquier directorio que se desee.

El módulo del mod_fastcgi de Oracle.

FastCGI es conceptual muy similar al CGI, con dos diferencias importantes:

Los procesos de FastCGI son persistentes: después de acabar una petición, esperan una nueva petición en vez de salir.

En vez de usar variables de entorno y las pipas del sistema operativo, el protocolo de FastCGI multiplexa la información del ambiente, la entrada estándar, la salida, y los errores

sobre una sola conexión full-duplex. Esto permite que los programas de FastCGI funcionen en las máquinas remotas, usando conexiones del TCP entre el web server y el uso de FastCGI.

6.3 Configuración del módulo del mod_perl de Oracle.

La integración de un intérprete completo del Perl permite que el servidor del HTTP del Oracle funcione con las escrituras del cgi del Perl sin cargar a un intérprete fresco cada vez.

Jefes del análisis y salida inseparada: a diferencia del mod_cgi, el mod_perl no hace ningún trabajo adicional que calcula los jefes que son enviados generalmente por las escrituras ordinarias del cgi debido a las acciones de otros módulos. Para hacer que el mod_perl trabaje mejor mod_cgi, se puede utilizar el directorio de PerlSendHeader. Sin este directorio, la salida de mod_perl-generated el contenido se asemeja a a nonparsed la escritura del jefe.

Mod_perl de incialización en el arranque: Con el mod_perl, se pueden cargar los módulos cuando el servidor del HTTP de Oracle empieza a correr. Esto tiene dos ventajas. Primero, significa que el módulo no tiene que ser cargado cuando otro módulo o escritura del cgi que funciona en Apache lo solicite. En segundo lugar, hace el módulo disponible como recurso común para todos los módulos y escrituras para su uso.

Reinicio del mod_perl cuando el servidor del http. Generalmente, cuando se recomienza el servidor del HTTP de Oracle, el mod_perl conserva a todos los módulos disponibles como todas las escrituras del cgi registradas en Apache. Si esto no es lo que se desea, se puede forzar el mod_perl para recomenzar también usando PerlFreshRestart. Desafortunadamente, algunos módulos y las escrituras del cgi no dirigen esto muy bien y pueden hacer el servidor del HTTP caerse, así por default está apagado.

7. Autentificación, autorización, y control de acceso en el servidor http.

La autentificación es cualquier proceso por el cual se puede verificar que alguien sea quién dice ser. Esto implica generalmente un username y una contraseña, pero puede incluir cualquier otro método de demostrar identidad, tal como una tarjeta, una exploración de la retina, un reconocimiento de voz, o huellas digitales.

La autorización se encarga de permitir a la persona, una vez que esté identificada, para tener acceso al recurso. Esto es determinada generalmente descubriendo si esa persona parte de un grupo particular, si esa persona ha pagado la admisión, o tiene un nivel particular de la habilitación.

Finalmente, el control de acceso es una manera mucho más general de hablar del acceso a un recurso. El acceso se puede conceder o negar basado en una variedad amplia de criterios o políticas, tales como la dirección de red del cliente, de la hora, de la una frace particular o del browser que el visitante está utilizando. El control de acceso puede trabar las puertas en el tiempo de cierre. Esto está controlando la entrada por una cierta condición arbitraria que pueda o pueda tener cualquier acceso a ciertas cualidades en particular

Porque estas tres técnicas se relacionan tan de cerca en la mayoría de los usos de los recursos informáticos. En detalle, la autentificación y la autorización están, en la mayoría de los casos de la mano.

Si se tiene información sobre el Web Site que sea sensible, o previsto para solamente un grupo de gente pequeño, las técnicas en esta clase particular ayudan a cerciorarse de que la gente que ve esas páginas es la gente que tenga permitido verla.

Autentificación básica

El nombre implica, la autentificación básica, este es el método más simple de autentificación, y era durante mucho tiempo el método más común de la autentificación usado.

7.1 Cómo trabaja la autentificación básica.

Cuando se ha protegido un recurso particular usando la autentificación básica, Apache envía la autentificación que requirió el jefe con la respuesta a la petición, para notificar al cliente que las credenciales del usuario se deben proveer en la orden para que el recurso sea devuelto

Sobre la recepción de un jefe de una cierta cantidad de respuestas, el browser del cliente, si apoya la autentificación básica, pedirá que el usuario provea un username y una contraseña que se enviarán al servidor. Si se está utilizando un browser gráfico, tal como Internet Explorer de Netscape o, observaremos qué es una caja a la cual hace estallar para arriba y le da un lugar mecanografía adentro del username y contraseña, ser enviado de nuevo al servidor. Si el username está en la lista aprobada, y si la contraseña provista está correcta, el recurso será vuelto al cliente.

Porque el protocolo del HTTP es noble, cada petición será tratada de la misma manera, aunque son del mismo cliente. Es decir, cada recurso que se pregunta el servidor tendrá que proveer el excedente de las credenciales de la autentificación otra vez para recibir el recurso.

De modo que se tendrá que mecanografiar solamente adentro el username y contraseña una vez por sesión del browser, es decir, pudimos tener que mecanografiarla adentro otra vez y así mismo la próxima vez que abrimos el browser y visitamos el mismo Web site.

Junto con la respuesta, otra información será pasada de nuevo al cliente. Es decir, envía un nombre que se asocie al área protegida del Web site. Esto se llama el reino, o apenas el nombre de la autentificación. El browser del cliente deposita el username y la contraseña que se proveyó en los almacenes junto con el reino de la autentificación, de modo que si otros recursos le preguntan al mismo reino, el mismo username y la contraseña se puede volver para authenticar esa petición sin requerir al usuario mecanografiarlos adentro otra vez. Esto que deposita es generalmente necesario para la sesión actual del browser, pero algunos browsers permiten que se almacenen permanentemente, de modo que nunca se tenga que mecanografiar adentro la contraseña otra vez.

El nombre de la autentificación, o el reino, aparecerá en la caja pop-up, para identificar lo que se están solicitando el username y la contraseña.

7.1.1 Crear un archivo de la contraseña

Para determinarse si una combinación particular de username/password es válida, el username y la contraseña provista por el usuario necesitarán ser comparados a algún listado autoritario de usernames y de la contraseña. Éste es el archivo de la contraseña, que s necesitará crear en el lado del servidor, y puebla con los usuarios válidos y sus contraseñas.

Porque este archivo contiene la información sensible, debe ser almacenado fuera del directorio de documento. Aunque, las contraseñas se cifran en el archivo, si una cookie accediera al archivo, sería una ayuda en su tentativa de querer trabajar fuera de las contraseñas. Porque la gente tiende a ser descuidada con las contraseñas que ella elige, y utilice la misma contraseña para la autentificación del Web site que para su cuenta bancaria, ésta potencialmente sea una abertura muy seria de la seguridad, incluso si el contenido en el Web site no es particularmente sensible.

Una practica muy importante es animar a los usuarios que utilicen una diversa contraseña para el Web site que para otras cosas más esenciales. Por ejemplo, mucha gente tiende a utilizar dos contraseñas una para todos de sus cosas extremadamente importantes, tales como la conexión a la computadora de escritorio, y para su cuenta bancaria, y otra para las cosas menos sensibles, el compromiso de las cuales sería menos serio.

Para crear el archivo de la contraseña en un servidor HTTP, debemos utilizar la instrucción htpasswd que vino con Apache. Esto será situado en el directorio del compartimiento de donde quiera que se instale el servidor de apache. Por ejemplo, será probablemente localizado en /usr/local/apache/bin/htpasswd.

Para crear el archivo de contraseña, introducimos las siguientes lineas:

htpasswd - username de c/de usr/local/apache/passwd/passwords

el htpasswd pedirá la contraseña y después pide que ingresemos la misma contraseña para confirmarla:

htpasswd c/usr/local/apache/passwd/passwords rbowen nueva contraseña, el mypassword escribe de nuevo nueva contraseña a mano el mypassword agrega la contraseña para el usuario rbowen

El archivo de la contraseña que se está creando contiene a un usuario llamado rbowen , y este archivo de la contraseña se está colocando en la localización /usr/local/apache/passwd/passwords . Substituiremos la localización, y el username, que deseamos utilizar para comenzar el archivo de la contraseña.

Si el htpasswd no está en una trayectoria, tendremos que ingresar la ruta al archivo para conseguir su funcionalidad. Es decir substituiría por / usr/local/apache/bin/htpasswd del htpasswd

Además se utiliza una bandera de C solamente cuando se está creando un archivo nuevo. Después de la primera vez se omitirá dicha bandera de c, cuando estemos agregando nuevos usuarios a un archivo ya existente de la contraseña.

sungo del htpasswd/usr/local/apache/passwd/passwords

En el ejemplo, agregará a usuario nombrado sungo a un archivo de la contraseña la cuál se ha creado ya anteriormente. Como antes mencionado pedirán la contraseña en la línea de comando y después serán pedidos confirmar la contraseña mecanografiándola otra vez.

Como precaución debemos tener cuidado cuando agregamos a nuevos usuarios a un archivo existente de la contraseña donde nosotros no utilizamos bandera de c para marcar un error.

El usar la bandera de c creará un archivo nuevo de la contraseña, incluso si se tiene ya un archivo existente de ese nombre. Es decir, quitará el contenido del archivo que está allí, y lo substituirá por un archivo nuevo que contiene solamente el un username que nosotros agregamos

La contraseña se almacena en el archivo de la contraseña en forma cifrada, de modo que los usuarios en el sistema no puedan leer el archivo y determinar inmediatamente las contraseñas de todos los usuarios. Sin embargo, debemos almacenar el archivo de forma segura en una localización, con cualesquiera permisos mínimos en el archivo de modo que el web server sí mismo pueda leer el archivo. Como por ejemplo, si el servidor se configura para funcionar como usuario nogroup. Entonces debemos fijar permisos en el archivo de modo que solamente el webserver pueda leer el archivo y solamente la raíz pueda escribirle:

chown root.nogroup/chmod 640/usr/local/apache/passwd/passwords de
usr/local/apache/passwd/passwords

En Windows, una precaución similar se debe tomar, cambiando la propiedad del archivo de la contraseña al usuario del Web Server, de modo que otros usuarios no puedan leer el archivo.

Una vez que se haya creado el archivo de la contraseña, se necesita decirle a Apache sobre este archivo y apache puede utilizar este archivo para requerir las credenciales del usuario para la admisión. Esta configuración se hace con los directorios en la tabla 1.

AuthType	Tipo de la autentificación que es utilizado. En este caso, será fijado a básico
AuthName	El reino o el nombre de la autentificación
AuthUserFile	La localización del archivo de la contraseña
AuthGroupFile	La localización del archivo del grupo, si lo hay
Requiera	El requerimiento que se debe satisfacer para conceder la admisión

Tabla 1.Directorios requeridos para las contraseñas

Estos directorios se pueden poner en los htaccess y archivar en el directorio particular que es protegido, o pueden entrar en el archivo principal de la configuración del servidor, en una sección del directorio raíz o el otro envase del alcance.

En el siguiente ejemplo demostramos como se define un reino de la autentificación llamado por solo invitación. Este esta localizado en archivo/usr/local/apache/passwd/passwords de la contraseña será utilizado para verificar la identidad del usuario. Solamente los usuarios nombrados rbowen o el sungo será concedido el acceso, y lo iguala entonces solamente si proporcionan una contraseña que empareje la contraseña almacenada en el archivo de la contraseña.

```
AuthType Basic
AuthName "By Invitation Only"
AuthUserFile /usr/local/apache/passwd/passwords
Require user rbowen sungo
```

El cgi (interfaz de entrada común) define una manera para que un Web Server obre recíprocamente con los programas de contenido externos, que se refieren a menudo como programas del cgi o escrituras del cgi. Es el más simple y la mayoría el mas común, esta es una manera de poner el contenido dinámico en un Web Site.

7.1.2 Configuración de Apache para permitir el CGI.

Para conseguir que los programas del cgi trabajen correctamente, necesitaremos tener el servidor de Apache configurado para permitir la ejecución del CGI. Hay varias maneras de hacer esto. Para entender la interacción y funcionalidad de un CGI la siguiente imagen nos dará un panorama abierto de las tecnologías implicadas en los servicios Web.

8. Los servicios WEB

8. 1 Ventajas de los servicios WEB.

Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad Firewall sin necesidad de cambiar las reglas de filtrado.

8.2 Razones para crear los servicios WEB

La principal razón para usar servicios Web es que se basan en http sobre TCP en el puerto 80. Muchas empresas se protegen mediante firewall que filtran y bloquean gran parte del tráfico de Internet. Por ello se cierran casi todos los puertos salvo el 80, porque es el que usan los navegadores. Los servicios Web se realizan por este puerto y ello los hace muy convenientes.
Otra razón es que antes de que existieran los servicios de web no había buenas interfaces para acceder a las funcionalidades de otros ordenadores en red. Una tercera razón por la que los servicios Web son muy prácticos es que pueden aportar un débil acoplamiento entre una aplicación que usa el servicio Web y el propio servicio. De esta forma los cambios que cada uno realice con el tiempo no deben afectar al otro. Esta flexibilidad será cada vez más importante, dado que la tendencia a construir las aplicaciones grandes a partir de componentes distribuidos más pequeños es cada día mayor.

8.3 Inconvenientes de los servicios WEB

- Para realizar transacciones no pueden compararse en su grado de desarrollo con los estándares abiertos de computación distribuida.

- Su rendimiento es bajo si se compara con otros modelos de computación distribuida, tales como RMI, CORBA o DCOM. Es uno de los inconvenientes derivados de adoptar un formato basado en texto. Y es que entre los objetivos de XML no se encuentra la concisión ni la eficacia de procesamiento. Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en *firewall* cuyas reglas tratan de bloquear o auditar la comunicación entre programas a ambos lados de la barrera.

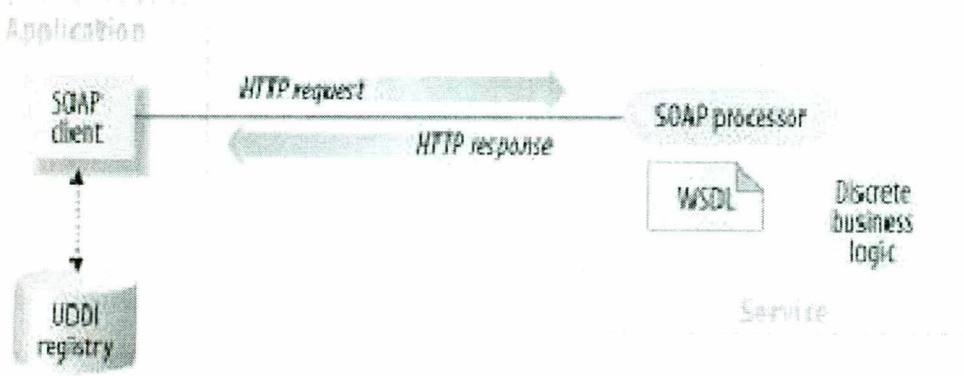


Figura 5. Tecnologías implicadas en los servicios WEB

8.4 ScriptAlias

En el ScriptAlias el directorio le dice a Apache que un directorio particular esté puesto a un lado para los programas del cgi. Apache asumirá que cada archivo en este directorio es un programa del cgi y procurará ejecutarlo, cuando ese recurso particular es solicitado por un cliente.

El directorio deScriptAlias:

// usr/local/apache/cgi-bin de ScriptAlias/del cgi-compartimiento/

Este ejemplo es si se realiza una instalación por default. El directorio ScriptAlias está como Alias, y este define un prefijo del URL que esté a tras a un directorio particular. Alias y ScriptAlias se utilizan generalmente para los directorios que están fuera de DocumentRoot directorio. La diferencia en medio Alias y ScriptAlias es eso ScriptAlias tiene el significado agregado que todo bajo ese prefijo del URL será considerado un programa del cgi. Así pues Apache toma como petición un recurso que comienza con /cgi usándolo en el directorio /usr/local/apache/cgi-bin/ y debe ser tratado como programa del cgi.

Por ejemplo, si el URL http://www.example.com/cgi-bin/test.pl hace una solicitud, Apache procurará ejecutar el archivo /usr/local/apache/cgi-bin/test.pl y devuelve la salida. Por supuesto, el archivo tendrá que existir y ser ejecutable y devuelta una salida de una manera particular o por el contrario Apache devolverá un mensaje de error.

8.5 Cgi fuera de los directorios de ScriptAlias

Los programas del cgi se restringen a menudo a ScriptAlias' por razones de la seguridad. De esta manera, los administradores pueden controlar firmemente quién se admite para utilizar programas del cgi. Sin embargo, si se toman las precauciones apropiadas de la seguridad, no hay razón por la que los programas del cgi no puedan funcionar de directorios arbitrarios. Por ejemplo, si se quiere dejar a los usuarios tener cierto contenido en los directorios caseros con UserDir directivo. Si desean tener sus propios programas del cgi, pero no tienen acceso a la programación del cgi, necesitarán poner a funcionar sus programas de cgi en otra parte.

Se podrían utilizar explícitamente opciones de directorio, dentro del archivo principal de la configuración del servidor, y especificar que la ejecución del cgi sea permitida en un directorio particular:

< directorio/usr/local/apache/htdocs/somedir > </directorio > de las opciones +ExecCGI

El directorio antes mencionado le permite a Apache la ejecución de los archivos del cgi. También necesitaremos decirle al servidor que son archivos del cgi. El directorio

AddHandler le dice al servidor como tratar todos los archivos con CGi o PL como programas del cgi:

8.6 Archivos htaccess

Los archivos htaccess son una manera de fijar directorios de la configuración sobre una base del directorio. Cuando Apache sirve un recurso, mira en el directorio de el cual está sirviendo un archivo para un archivo llamado htaccessy, si lo encuentra, aplicará los directorios encontrados en este. htaccess hace que los archivos se pueden permitir con el directorio AllowOverride, el cual especifica qué tipos de directorios pueden aparecer en estos archivos, o si no se permiten en todos.

Opciones De AllowOverride

En el archivo htaccess, necesitaremos el directorio siguiente:

Opciones +ExecCGI el cuál le dice a Apache que la ejecución de los programas del cgi esten permitidos en este directorio.

9. Como escribir un programa del cgi para HTTP

Primero, toda la salida del programa del cgi se debe preceder por un jefe. Este es el jefe del HTTP que le dice al cliente qué clase de contenido está recibiendo. La mayoría del tiempo, este será observado por:

Contenido-tipo: text/html

En segundo lugar, la salida necesita estar en el HTML u otro formato de tal forma que un browser pueda mostrar. La mayoría del tiempo, éste será HTML, pero podemos hacer que de vez en cuando escriba el programa del cgi el cual hace salir una imagen del GIF, o el otro contenido no HTML.

Incluso si no se es familiar con el Perl, debemos poder ver qué está sucediendo aquí. La primera línea dice Apache que este programa puede ser ejecutado alimentando el archivo al

intérprete encontrado en la localización /usr/bin/perl. La segunda línea imprime la declaración del contenido del tipo que hablamos, seguido por dos pares del newline de la vuelta de carro. Esto pone una línea en blanco después del jefe, para indicar el extremo de los jefes del HTTP y el principio del cuerpo. La tercera línea imprime la secuencia hola, mundo.

Si ahora abrimos el browser preferido e introducimos la siguiente dirección: http://www.example.com/cgi-bin/first.pl, o dondequiera que se ponga el archivo, veremos la línea Hola, Mundo que aparecen en la ventana del browser. No es muy emocionante, pero una vez que se consiga eso que trabaje, tendremos un buen ejemplo

9.1 Errores de sintaxis

La mayoría del tiempo cuando un programa del cgi falla, está debido a un problema con el programa mismo. Esto es particularmente verdad una vez que se consiga el dominio de esta materia, e incurre en no más de una las dos equivocaciones antedichas. Debemos procurar siempre probar el programa en la línea de comando antes de hacerlo vía browser. Esto eliminará la mayoría de los problemas.

9.2 Registros de errores

Los registros de errores debemos de tomarlos como nuestros files amigos. Cualquier cosa que va mal genera un mensaje en el registro de errores. Debemos observar primeramente el registro de errores antes de cualquier otra decisión a tomar. Si el lugar en donde se está recibiendo el Web Site no permite que se tenga acceso al registro de errores, se debe recibir probablemente el sitio en alguna otra parte. Tenemos que aprender a leer los registros de errores y encontraremos que casi todos los problemas serán identificados rápidamente y solucionado rápidamente.

9.3 Compilación de Apache

Si desea usar la interfaz estilo autoconf, deberá leer el fichero INSTALL en el directorio raíz de la distribución fuente de Apache. Para la compilación e instalación en plataformas específicas podemos apoyarnos de las siguientes bibliografías:

Usar Apache con Microsoft Windows
Usar Apache con Cygwin
Usar Apache con Novell Netware 5
Usar Apache con HP MPE/iX
Compilación de Apache bajo UnixWare
Vistazo general de la versión TPF de Apache

La compilación de Apache consiste en tres pasos. Primero seleccionar qué módulos de Apache se desean incluir en el servidor. Segundo crear una configuración para el sistema operativo. Tercero compilar el ejecutable.

Toda la configuración de Apache está en el directorio src de la distribución.

Seleccionar los módulos para compilar, en el fichero de configuración de Apache. Descomentar las líneas correspondientes a los módulos opcionales que se desean incluir (entre las líneas AddModule al final del fichero), o escribir nuevas líneas correspondientes a módulos adicionales que se hayan bajado o programado. Los usuarios avanzados pueden comentar los módulos por default si están seguros de que no los necesitan aunque hay que tener cuidado, ya que algunos de estos módulos son necesarios para el buen funcionamiento y una correcta seguridad del servidor.

Es recomendable leer también las instrucciones del fichero de Configuración para comprobar si se necesita configurar otras líneas.

Configurar Apache para el sistema operativo. Se puede ejecutar un script como el mostrado más abajo. Aunque si esto falla o se tiene algún requerimiento especial como por ejemplo incluir una librería adicional exigida por un módulo opcional, podemos editarlo para

utilizar en el fichero de Configuración las siguientes opciones: EXTRA_CFLAGS, LIBS, LDFLAGS,INCLUDES.

Ejecutar el script de configuración:

```
% Configure
Using 'Configuration' as config file
 + configured for <whatever> platform
 + setting C compiler to <whatever> *
 + setting C compiler optimization-level to <whatever> *
 + Adding selected modules
 + doing sanity check on compiler and options
Creating Makefile in support
Creating Makefile in main
Creating Makefile in os/unix
Creating Makefile in modules/standard
```

Obviamente dependiendo de la configuración y del sistema operativo. El resultado podría no coincidir con el mostrado.

Esto genera un fichero Makefile a ser usado en el tercer paso. También crea un Makefile en el directorio support, para la compilación de programas de soporte.Si se desean mantener varias configuraciones, se puede indicar a Configure una de las opciones en un fichero.

Los módulos de la distribución de Apache son aquellos que se han probado y utilizado regularmente por varios miembros de un grupo de desarrollo de Apache. Los módulos adicionales creados por miembros del grupo o por terceras personas para necesidades o funciones específicas están disponibles en:

http://www.apache.org/dist/httpd/contrib/modules/.

Se tendrá un fichero binario llamado hhtpd en el directorio src. Una distribución binaria de Apache ya traerá este fichero.

El próximo paso es instalar el programa y configurarlo. Apache esta diseñado para ser configurado y ejecutado desde los directorios donde fue compilado. Si se quiere ejecutarlo desde otro lugar, se debe crear un directorio y copiar los directorios conf, logs e icons. En cualquier caso se deberán leer las sugerencias de seguridad que describen cómo poner los permisos del directorio raíz.

El paso siguiente es editar los ficheros de configuración del servidor. Consiste en configurar varias directivas en los tres ficheros principales. Por default, estos ficheros están en el directorio conf y se llaman srm.conf, access.conf y httpd.conf. Para comenzar, hay ejemplos de estos ficheros en el directorio de la distribución, llamados srm.conf-dist, access.conf-dist y httpd.conf-dist. Se pueden copiar o renombrar estos ficheros a los correspondientes nombres sin la terminación dist. Hay que editar cada uno de ellos. Debemos leer los comentarios cuidadosamente. Un error en la configuración de estos ficheros podría provocar fallos en el servidor o volverlo inseguro. Tenemos también un fichero adicional en el directorio conf llamado mime.conf. Este fichero normalmente no tiene que ser editado.

Primero editamos el fichero http.conf. Este configura atributos generales del servidor como el número de puerto, el usuario que lo ejecuta, etc. El siguiente a editar es srm.conf; este fichero configura la raíz del árbol de los documentos, funciones especiales como HTML analizado sintácticamente por el servidor, mapa de imagen, etc. Finalmente, editamos access.conf que configura los accesos.

Además de estos tres ficheros, el comportamiento del servidor puede ser modificado directorio a directorio usando los ficheros .htaccess para los directorios en los que acceda el servidor.

9.4 Configurando el sistema de tiempo correctamente

Una operación de un servidor web requiere un tiempo concreto, ya que algunos elementos del protocolo HTTP se expresan en función de la hora y el día. Por eso es importante saber que tipo de configuración NTP o de otro sistema de sincronización de un Unix o lo que haga de equivalente en Windows.

9.5 Programas de soporte para la compilación

Además del servidor principal httpd que se compila y configura como hemos visto, Apache incluye programas de soporte. Estos no son compilados por default. Los programas de soporte están en el directorio support. Para compilar esos programas, tenemos que entrar en el directorio indicado y ejecutar el comando correspondiente.

En Windows, Apache funciona normalmente como servicio en Windows NT, o como uso de la consola en Windows 95. Esto no se aplica en su lleno extiende para la plataforma de Cygwin.

En Unix, funciona el programa del httpd mientras que un demonio que se ejecute continuamente en el fondo para manejar peticiones. Es posible hacer que Apache sea invocado por el demonio del Internet inetd una conexión al servicio del HTTP se hace cada vez usando el directorio de ServerType, pero esto no se recomienda.

Si el puerto especificado en el archivo de la configuración es por default el 80 o cualquier otro puerto debajo de 1024, entonces es necesario tener privilegios de la raíz para comenzar Apache, de modo que pueda atar a este puerto privilegiado. Una vez que el servidor haya comenzado y haya terminado algunas actividades preliminares tales como abrir sus ficheros de diario, lanzará varios procesos los cuales haga el trabajo de escuchar y contestando preguntas a los clientes. El proceso httpd continúa funcionando como el usuario de la raíz, pero los procesos del niño funcionados como usuario menos privilegiado. Esto es controlado por los directorios de proceso de la creación de Apache.

httpd cuando se invoca debe localizar y leer el archivo de la configuración httpd.conf. La localización de este archivo se fija en tiempo de compilación, pero es posible especificar su localización usando el tiempo de pasada.

/usr/local/apache/bin/httpd - f/usr/local/apache/conf/httpd.conf

Como alternativa a la invocación httpd binario directamente, un shell script llamado el apachectl se proporciona tales como el cual puede ser utilizado y controlar el proceso del demonio con comandos simples comienzo del apachectl y parada del apachectl.

Si todo va bien durante el arranque, el servidor separará del Terminal y el aviso de comando volverá casi inmediatamente. Esto indica que el servidor esta en servicio. Podemos entonces utilizar el browser para conectar con el servidor y para observar la página de prueba, el directorio de DocumentRoot y la copia local de la documentación ligada de esa página.

9.6 Errores Durante Start Up

Si Apache sufre un problema durante el arranque, escribirá un mensaje que describe el problema a la consola o al ErrorLog antes de salir. Uno de los mensajes de error más comunes es "Incapaz atar para virar hacia el lado de babor". Este mensaje es causado generalmente por cualquiera de los siguientes casos:

El intentar encender el servidor en un puerto privilegiado cuando no está entrado como el usuario de la raíz.
El intentar encender el servidor de cuando hay otro proceso de Apache o un Web Server limitado ya al mismo puerto.

9.7 El comenzar en el Cargador del Tiempo

Si se desea que el servidor continuara funcionando después de que un reboot del sistema, se debe agregar una llamada a httpd o apachectl a sus archivos de arranque de sistema (típicamente rc.local o un archivo en rc.N directorio). Esto comenzará Apache como raíz. Antes de hacer esto aseguremos de que el servidor está configurado correctamente para las restricciones de la seguridad y del acceso. En apachectl la escritura se diseña para poderla ligarse a menudo directamente como escritura del init, para que sea seguro comprobar los requisitos exactos del sistema.

9.8 Información Adicional

La información adicional sobre las opciones del comando línea del httpd y del apachectl así como otros programas de ayuda que están incluidos con el servidor y disponible en el servidor y la página de los programas de soporte. Hay también documentación en todos los módulos incluidos con la distribución de Apache y los directorios que proporcionan.

10. Usar Apache Con Microsoft Windows

Apache en el NT todavía no se ha optimizado para el funcionamiento. Apache todavía se realiza lo mejor posible, y es el más confiable en las plataformas de Unix. El funcionamiento en un cierto plazo del NT ha mejorado, y el gran progreso se está haciendo en las versiones próximas de Apache para las plataformas de Windows. Todavía piden los usuarios hacer revisiones comparativas del funcionamiento del webserver contra Apache en una plataforma de Unix tal como Solaris, FreeBSD, o Linux.

10.1 Requisitos

Apache se diseña para funcionar en Windows NT 4,0 y Windows 2000, 2003, 2003 Server. El instalador binario trabajará solamente con la familia x86 de procesadores, tales como Intel. Apache puede también funcionar en Windows 95 y 98, pero éstos no se han probado. En todos los casos el protocolo TCP/IP debe ser instalado.

Si funciona en NT 4.0 instalando los servicios correspondientes como ediciones creadas del paquete 4 del servicio con integridad de TCP/IP y del Winsock que fueron resueltas en el paquete 5 del servicio. Winsock2 se requiere para Apache 1.3.7 en adelante.

Si funciona en Windows 95, la mejora Winsock2 debe ser instalada antes de que Apache funcione. Winsock2 para Windows 95 está disponible en las versiones recientes de apache. Hay que comentar que el establecimiento de una red de dialup 1.2 y sus actualizaciones incluyen un Winsock2 y la actualización Winsock2 debe ser reinstalada después de instalar

una red del dialup de Windows 95. Windows 98, el NT y 2000, los usuarios necesitan no tomar ninguna acción especial, esas versiones proporcionan Winsock2 según lo distribuido.

10.2 Descargar Apache para Windows

La información sobre la versión más reciente de Apache se puede encontrar en el Web Server de Apache (http://httpd.apache.org/). Esto enumerará el lanzamiento actual, los lanzamientos más recientes de las pruebas alfa o de la beta.

Se debe descargar la estructura binaria de Apache para Windows nombrado como apache - win32-src.msi si nos encontramos en el código de fuente, o simplemente interesados en apache #-win32-no_src.msi. Si no planeamos hacer cualquier cosa con el código de fuente y apreciar un rápido descargue. Cada uno de estos archivos contiene el tiempo de pasada completo de Apache. Debemos tener la versión 1.10 del instalador de Microsoft instalado en la PC antes de que podamos instalar las distribuciones del tiempo de bypass de Apache. Windows 2000 es entregado con la ayuda del instalador de Microsoft, otras necesitarán descargarla. Para más información, podemos visitar la página principal en http://httpd.apache.org/download.cgi. Las instrucciones en la localización del instalador de Microsoft, así como las distribuciones binarias de Apache, se encuentran en el directorio de la transferencia directa win32.

Con un servidor de transacciones, el cliente invoca procedimientos remotos ó servicios ellos residen en el servidor el cual contiene el motor de la base de datos SQL. Estos procedimientos remotos en el servidor ejecutan un grupo de tareas SQL. El intercambio en la red consiste de un mensaje request/reply opuesto al anterior donde existe un mensaje request/reply por cada transacción SQL. Estos requerimientos SQL agrupados, se denominan transacciones. Con un servidor de transacciones, se debe crear la aplicación cliente/servidor para ambos componentes. El componente para el cliente usualmente incluye una interfase gráfica de usuario, GUI. El componente en el servidor consiste del servidor de transacciones de la base de datos. Este tipo de aplicaciones se denominan Online Transaction Processing ó OLPT. Las OLPT requieren alto control de seguridad e integridad de los datos, dos formas de OLPT son TP-Lite, basado en la carga de

procedimientos provistos por los vendedores de bases de datos y TP-Heavy basado en monitores TP. (TP = Transaction Processing Monitors,) especializados en administrar transacciones desde el punto origen típicamente el cliente a través de uno o más servidores y retornar el resultado al cliente. Cuando una transacción finaliza los monitores TP aseguran que todos los sistemas envueltos en la transacción se encuentran en un estado consistente, ellos saben como ejecutar la transacción, la ruta de las transacciones a través de todo el sistema, el balance de carga para su ejecución y como restaurarla en caso de fallas.

El código fuente está disponible en el directorio src.msi, o del directorio de la distribución como un archivo Zip. Si se desea realizar la compilación de Apache por uno mismo, no hay necesidad de instalar tampoco paquete msi. El archivo zip contiene solamente el código de fuente, con las conclusiones de la línea del MSDOS, estas son conclusiones de la línea de cr/lf, en vez del solo lf usado para los archivos de Unix distribuidos en archivos del tar.gz o del tar.Z.

Mientras que la fuente está también disponible como el archivo llamado tar.gz tar.Z, éstos contienen las conclusiones de la línea de unix lf que causan los problemas para los usuarios de Windows. Para utilizar esos archivos, se deben convertir por lo menos mak y dsp los archivos para tener conclusiones de la línea del DOS antes de MSVC.

10.3 Instalación de Apache para Windows

Si se desea instalar el servicio de apache para todos los usuario o solamente ponerlo disponible como servicio en un servidor de apache y manejarlo a través de una consola.
El nombre del servidor, Domain Name y una cuenta administrativa de correo.
El directorio para instalar Apache por default es C:\Program Files\Apache Group\Apache aunque se puede cambiar a cualquier otro directorio si se desea.

El tipo de la instalación. La opción "completa" instala todo, incluyendo el código de fuente si es que se descargo el paquete src.msi. Si elegimos el personalizado podemos elegir no instalar la documentación, o el código de fuente de ese paquete.

Durante la instalación, Apache configurará los archivos en el directorio conf el cual es un directorio elegido de la instalación. Sin embargo si existe cualquiera de los archivos en este directorio ya no serán sobrescritos. En lugar la nueva copia del archivo correspondiente será dejada con la extensión default.conf. Así pues, por ejemplo, si conf\httpd.conf existe ya no será alterado, pero la versión que habría sido instalada será dejada adentro de conf\httpd.default.conf. Después de que la instalación haya acabado se debe comprobar manualmente para ver lo que se le instalo al archivo default.conf, y en caso de necesidad poner al día los archivos existentes de la configuración.

También, si creamos archivo llamado htdocs\index.html entonces no será sobrescrito (no index.html.default. Esto debe significar que es seguro instalar Apache sobre una instalación existente solamente se tendrá que parar el funcionamiento del servidor existente antes de hacer la instalación e inmediatamente después de la instalación arranca nuevamente. Después de instalar Apache, se deben corregir los archivos de la configuración en conf directorio según lo requerido. Estos archivos serán configurados durante la instalación lista para ser ejecutados por Apache en el directorio donde fue instalado, con los documentos servidos del subdirectory htdocs. Hay porciones de otras opciones que deban ser fijadas antes de que se comience realmente a usar Apache.

Si la instrucción uninstall Apache, no llegaran a quitar la configuración y los ficheros de diario. Necesitaremos suprimir el árbol del directorio de la instalación (grupo de C:\Program Files\Apache por default) si no tenemos el cuidado de guardar la configuración y otros archivos. Puesto que el archivo de httpd.conf es su esfuerzo acumulado al usar Apache, necesitamos hacer el esfuerzo de quitarlo. Igual sucede para el resto de los archivos que se pudieron haber creado, tan bien como cualquier fichero de diario Apache.

10.4 Archivos Principales de la Configuración

Apache es configurado poniendo directorios en archivos llanos de la configuración del texto. El archivo principal de la configuración se llama generalmente httpd.conf. La localización de este archivo se fija en de tiempo de compilación, pero se puede eliminar con una bandera de la línea de comando. Algunos sitios también tienen srm.conf y access.conf

por razones históricas. Además, otros archivos de la configuración se pueden usar agregado directorio incluya. El directorio se puede poner en cualquiera de estos archivos de la configuración. Los cambios a los archivos principales de la configuración son reconocidos solamente por Apache cuando se comienza o se reinicia.

Una característica de apache es si cualquier archivo de la configuración es realmente un directorio, Apache incorporará ese directorio y analizará cualquier archivo y subdirectorios encontrados allí como archivos de la configuración. Un uso posible para esto sería agregar VirtualHosts creando los archivos pequeños de la configuración para cada anfitrión y colocándolos en un cierto directorio de la configuración. Así, podemos quitar VirtualHosts sin corregir ningún archivo, simplemente la edición o suprimirlos. Esta automatización hace los procesos mucho más fácil.

El servidor también lee un archivo que contiene tipos del documento del MIME; el nombre de fichero es fijado por el directorio de TypesConfig, y es mime.types por default.

Los archivos de la configuración de Apache contienen un directorio por línea. "\" del back-slash se puede utilizar como el carácter pasado en una línea para indicar que el directorio continúa sobre la línea siguiente. No debe haber otros caracteres o espacio blanco entre el back-slash y el extremo de la línea.

Los directorios en los archivos de la configuración son insensibles, pero las discusiones a los directorios son a menudo caso sensible. Las líneas que comienzan con el carácter "#" se consideran los comentarios, y no se le hacen caso. Los comentarios pueden no ser incluidos en una línea después de un directorio de la configuración. Las líneas en blanco y el espacio blanco que ocurre antes de que no haga caso un directorio, así que podremos señalar los directorios para la claridad.

Podemos comprobar los archivos de la configuración para saber si hay errores de sintaxis sin encender el servidor usando apachectl ó configtest opción de la línea de comando.

Las líneas siguientes demuestran una configuración completa para el servidor del HTTP de Oracle. Contiene todos los elementos y cualidades posibles de la configuración para el servidor del HTTP de Oracle.

```
<ias-component id="HTTP_Server" status="enabled" id-matching="false">
<process-type id="HTTP_Server" module-id="OHS">
<process-set id="HTTP_Server" restart-on-death="true" numprocs=1>
<module-data>
<category id="start-parameters">
<data id="config-file" value="/myconfs/httpd.conf"/>
<data id="start-mode" value="ssl-disabled"/>
<data id="command-line" value="-D MyDefine"/>
</category>
<category id="ping-parameters">
<data id="ping-url" value="/"/>
</category>
<category id="restart-parameters">
<data id="reverseping-timeout" value="345"/>
<data id="no-reverseping-failed-ping-limit" value="3"/>
<data id="reverseping-failed-ping-limit" value="6"/>
</category>
</module-data>
```

<start timeout="300" retry="3"/>

<stop timeout="300"/>

<restart timeout="300"/>

<ping timeout="30" interval="30"/>

</process-set>

</process-type>

</ias-component>

10.5 Los módulos

Se dice que apache es un servidor modular (tabla 2). Esto implica que solamente la funcionalidad más básica está incluida en el servidor de la base. Las características extendidas están disponibles a través de los módulos que se pueden cargar en Apache. Por default, un sistema bajo de módulos es incluido en el servidor en el tiempo de compilación. Si el servidor se compila para utilizar los módulos cargables dinámicamente, después los módulos se pueden compilar por separado y agregar en cualquier momento usando el directorio de LoadModule. Si no, Apache debe ser recompiled para agregar o para quitar los módulos. Los directorios de la configuración pueden ser condicionales incluido en una presencia de un módulo particular incluyéndolos en un bloque de IfModule .

Para ver qué módulos se compilan actualmente en el servidor, se pueden utilizar las opciones de las líneas de comando.

Módulos Relacionados	Directorios Relacionados
mod_so	AddModule ClearModuleList IfModul LoadModule

Apache es un servidor modular
Tabla 2. Los módulos de apache

10.6 Alcance de los directorios

Los directorios puestos en los archivos principales de la configuración (tabla 3) se aplican al servidor entero. Si se desea cambiar la configuración para solamente una pieza del servidor, se pueden alcanzar los directorios colocándolos adentro <Directorio>, <DirectoryMatch> , <Archivos> , < FilesMatch> , < Localización > y <LocationMatch> secciones. Estas secciones limitan el uso de los directorios que incluyen a las localizaciones particulares o a URLs del filesystem. Pueden también ser jerarquizadas, teniendo en cuenta la configuración muy fina.

Directorios Relacionados
directorio
DirectoryMatch
archivo
FilesMatch
localización
LocationMatch
VirtualHost

Directorios Relacionados
Tabla 3. Los directorios de Apache

Apache tiene la capacidad para servir muchos y diversos websites simultáneamente. Esto se llama Virtual Hosting . Los directorios pueden también estar en scoped colocándolos adentro <VirtualHost>, de modo que se apliquen solamente a los pedidos un website en particular.

Aunque la mayoría de los directorios se pueden poner en cualesquiera de estas secciones, algunos directorios no tienen sentido para algunos contextos. Por ejemplo, los directorios

que controlan la creación de proceso se pueden poner solamente en el contexto principal del servidor. Para encontrar qué directorios pueden ser puestos en los cuales las secciones, comprueban el contexto del directorio. Para la información adicional, proporcionamos los detalles en cómo las secciones del directorio, de la localización y de archivos trabajan.

Apache permite la gerencia descentralizada de la configuración vía los ficheros especiales colocados dentro del árbol de jerarquías. Los ficheros especiales se llaman generalmente htaccess solamente cualquier nombre se puede especificar en AccessFileName. Los directorios puestos adentro de htaccess y los archivos se aplican al directorio donde se coloca el archivo y a todos los subdirectorios. Los archivos de htaccess siguen la misma sintaxis que los archivos principales de la configuración. Desde entonces los archivos htaccess se leen en cada petición, cambios realizados en efecto inmediatos de la toma de estos archivos.

Para encontrar qué directorios se pueden poner dentro de los archivos htaccess, se comprueba el contexto del directorio. Los controles posteriores del administrador del servidor qué directorios se pueden poner adentro de los archivos htaccess configurando el directorio AllowOverride en los archivos principales de la configuración.

11. La Negociación en Apache

Para negociar un recurso, el servidor necesita tener la información sobre cada una de las variantes. Esto se hace de dos maneras:

Usar un mapa del tipo es decir, nombrar los archivos que contienen las variantes explícitamente.
Usando la búsqueda de un MultiViews, donde el servidor hace un fósforo implícito del patrón del nombre de fichero y elige entre de los resultados.

11.1 Usar un archivo de tipo mapa

Un mapa del tipo es un documento que se asocia al tratante nombrado tipo-mapa para la compatibilidad con más configuraciones pasadas de Apache, el tipo del MIME utiliza esta característica, se debe tener un tratante fijo en la configuración que define un sufijo del archivo como tipo-mapa éste es mejor hecho con: var del tipo-mapa de AddHandler en el archivo de la configuración del servidor.

Los archivos del mapa del tipo tienen una entrada para cada variante disponible; estas entradas consisten en líneas contiguas del jefe del formato HTTP. Las entradas para diversas variantes son separadas por las líneas en blanco. Las líneas en blanco son ilegales dentro de una entrada. Es convencional comenzar un archivo del mapa con una entrada para la entidad combinada en su totalidad aunque esto no se requiere, y si el presente es un caso no hecho. El archivo será nombrado foo.var y se coloca en el mismo directorio con las varias variantes del recurso foo.

Si las variantes tienen diversas calidades de la fuente, eso se puede indicarse por el parámetro de los "qs" al tipo de medios.

Los valores de los "qs" pueden variar en la gama 0,000 a 1,000. Observamos que cualquier variante con un valor de los qs de 0,000 nunca será elegida. Las variantes sin el valor de parámetro de los "qs" se dan un factor de los "qs" de 1.0.

El parámetro de los "qs" indica la calidad relativa de esta variante comparada a las otras variantes disponibles independientes de las capacidades del cliente. Por ejemplo, un archivo JPEG está generalmente de una calidad más alta de la fuente que un archivo ASCII si está procurando representar una fotografía. Sin embargo, si el recurso que es representado es un archivo original del ASCII, después una representación del ASCII tendría una calidad más alta de la fuente que una representación del JPEG. Un valor de los "qs" es por lo tanto específico a una variante dada dependiendo de la naturaleza del recurso que representa.

La lista completa de los jefes reconocidos es uri del archivo que contiene la variante (del tipo de medios dado, codificada con la codificación dada del contenido). Estos se interpretan como URLs concerniente al archivo del mapa; deben estar en el mismo servidor, y deben referir a los archivos a los cuales concedería el cliente el acceso si se solicitaran directamente.

Contenido tipo de medios de los parámetros del charset, del nivel y de los "qs" pueden ser dados. Éstos se refieren a menudo como tipos del MIME; los tipos de medios típicos son image/gif text/plaino text/html.

Contenido-Lengua:

Los idiomas de la variante, especificadas como etiqueta de lengua de estándar del Internet de RFC 1766.

Contenido-Codificación:

Si se comprime, o se codifica de otra manera, más bien que se contiene en el archivo las informaciones en bruto reales, este dice cómo fue hecha. Apache reconoce solamente los encodings que son definidos por un directorio de AddEncoding. Esto incluye normalmente los encodings comprimidos para los archivos del compress'd y x-gzip para los archivos del gzip'd. El prefijo no hace caso para las comparaciones de codificación.

Contenido-Longitud:

El tamaño del archivo. Especificar longitudes exactas, en el tipo-mapa permite que el servidor compare tamaños de los archivo sin la comprobación de los archivos reales.

Descripción:

Una descripción textual human-readable de la variante. Si Apache no puede encontrar ninguna variante apropiada para volver, volverá una respuesta de error que enumere todas

las variantes disponibles en lugar de otro. Una lista tan variable incluirá las descripciones variables human-readable.

11.2 Multiviews

MultiViews es una opción del directorio, significando que puede ser fijada con opciones de directorio dentro de < directorio > < localización > o < archivo > en la sección access.confo, si AllowOverride se fija correctamente, ó htaccess archivos. Obsérvese que esas opciones MultiViewsusted tiene que pedir él nombre.

El efecto de MultiViews es si el servidor recibe una petición para /some/dir/foosi /some/dir tiene MultiViews y /some/dir/foo no existe, entonces el servidor lee el directorio que busca los archivos nombrados foo y con eficacia falsifica sobre un mapa del tipo que nombre todos esos archivos, asignándoles los mismos tipos y contenido content-encodings de medios que tendría si el cliente había pedido uno de ellos por nombre. Entonces elige el mejor a los requisitos del cliente.

MultiViews puede también aplicarse a las búsquedas para el archivo nombrado por DirectoryIndex, si el servidor está intentando poner en un índice un directorio. Si los archivos de la configuración especifican el índice de DirectoryIndex entonces el servidor arbitrará en medio index.html index.html3 si ambos están presentes. Si ni unos ni otro están presentes el servidor lo funcionará.

Si uno de los archivos encontró cuando leía que el directorio es una escritura del cgi, aunque no es muy obvio que suceda, el código da a ese caso el tratamiento especial si la petición era un POSTE, o un CONSEGUIR con QUERY_ARGS o PATH_INFO, la escritura se le da un grado de la calidad extremadamente alta y se invoca generalmente si no se da un grado de la calidad extremadamente baja, que hace generalmente una de las otras opiniones para ser recuperado.

11.3 Los Métodos de la Negociación. Después de que Apache haya obtenido una lista de las variantes para un recurso dado, de un archivo del tipo-mapa o de los nombres de fichero en el directorio, invoca uno de dos métodos para decidir sobre la mejor variante a devolver. No es necesario saber cualesquiera de los detalles de cómo la negociación ocurre realmente para utilizar las características de la negociación de Apache.

11.3.1 Existen dos métodos de la negociación. La negociación conducida del servidor con el algoritmo de Apache se utiliza en el caso normal. Cuando se utiliza este algoritmo, Apache puede a veces conseguir el factor de calidad de una dimensión particular para alcanzar un resultado mejor (tabla 4). Se utiliza la negociación transparente cuando el browser solicita específicamente esto a través del mecanismo definido en RFC 2295. Este método de la negociación da al browser control completo sobre decidir la mejor variante, el resultado es por lo tanto dependiente en los algoritmos específicos usados por el browser. Como parte del proceso transparente de la negociación, el browser puede pedir que Apache funcione el correctamente.

Dimensión	Notas
Tipo De Medios	El browser indica preferencias con el campo del jefe del aceptado. Cada artículo puede tener un factor de calidad asociado. La descripción variable puede también tener un factor de calidad.
Lengua	El browser indica preferencias con el campo del jefe. Cada artículo puede tener un factor de calidad. Las variantes se pueden asociar a ningunos, uno o más que una lengua.
Codificación	El browser indica preferencia con el campo de Aceptar-Codificación del jefe. Cada artículo puede tener un factor de calidad.
Charset	El browser indica preferencia con el campo del jefe del Aceptado-Charset. Cada artículo puede tener un factor de calidad. Las variantes pueden indicar un charset como un parámetro del tipo de medios.

Tabla 4. Dimensión de la negociación de Apache

11.4 Algoritmo De la Negociación De Apache

Apache puede utilizar el algoritmo siguiente para seleccionar la mejor variante para devolver al browser una respuesta. Este algoritmo no es configurable adicional. Funciona de la siguiente manera:

1. Primero, para saber si hay dimensión de la negociación, comprobar si se aceptan el apropiado campo del jefe y asignan una calidad a cada variante. Si el jefe del aceptado para cualquier dimensión implica que esta variante no es aceptable, elimínela. Si sigue habiendo ningunas variantes, vaya al paso 4.

2. Seleccionar la mejor variante por un proceso de la eliminación. Cada una de las pruebas siguientes se aplica en orden. Cualquier variante no seleccionada en cada prueba se elimina. Después de que cada prueba, si sigue habiendo solamente una variante, la seleccione como la mejor opción y proceda al paso 3. Si sigue habiendo más de una variante, moverse encendido a la prueba siguiente.
Multiplique el factor de calidad del jefe del aceptado con el factor de la calidad de fuente para el tipo de medios de esta variante, y seleccione las variantes con el valor más alto.
Seleccione las variantes con el factor de calidad más alto de la lengua.
Seleccione las variantes con la mejor opción de la lengua, usando o la orden de idiomas en el jefe de la Aceptar-Lengua, o bien la orden de idiomas en el directorio de LanguagePriority .
Seleccione las variantes con el parámetro "llano" más alto de los medios (usado para dar la versión de los tipos de medios del texto/HTML).
Seleccione las variantes con los mejores parámetros de los medios del charset, según lo dado en la línea del jefe del Aceptado-Charset. Es aceptable a menos que esté excluido explícitamente. Variantes con un texto y el tipo de medios pero asociado no explícitamente a un charset particular.
Seleccione esas variantes que han asociado los parámetros de los medios del charset que no son ISO-8859-1. Si no hay tales variantes, seleccione todas las variantes en lugar de otro.

Seleccionar las variantes con la mejor codificación. Si hay variantes con una codificación que sea aceptable al usuario-agente, seleccionar solamente estas variantes. Si no, si hay una mezcla de variantes codificadas y no codificadas, seleccionar solamente las variantes unencoded. Si o se codifican todas las variantes o todas las variantes no se codifican, seleccione todas las variantes.

Seleccione las variantes con la longitud más pequeña.

Seleccione la primera variante de esos restantes. Este será el primer enumerado en el archivo del tipo-mapa, o cuando las variantes se leen en el directorio, el nombre del archivo primero cuando está clasificado usando orden del código del ASCII.

3. El algoritmo ahora ha seleccionado uno la mejor posible variante, así que devuelva como la respuesta. El jefe de la respuesta del HTTP varía se fija para indicar las dimensiones de la negociación (los browsers pueden utilizar esta información al depositar el recurso). Extremo.

4. Conseguir aquí significa que no se seleccionara ninguna variante (porque ninguna es aceptable al browser). Devuelve un estado de ninguna representación aceptable con un cuerpo de la respuesta que consiste en un documento del HTML que enumera las variantes disponibles. También fijar el http del jefe para indicar las dimensiones de la variación.

Se debe estar enterado que el mensaje de error vuelto por Apache es necesariamente algo conciso y puede ser que confunda a algunos usuarios, aun cuando enumera los alternativas disponibles. Si se desea evitar a usuarios que ven esta página del error, se deben organizar los documentos tales que un documento en un lenguaje por default está vuelto siempre si un documento no está disponible en las codificaciones e idiomas que el browser pidió.

Particularmente, si se quisiera que un documento en un lenguaje por default fuera devuelto y si un documento no está disponible en uno de los idiomas que el browser pedido, se debe crear un documento sin sistema de la cualidad del lenguaje.

Conclusiones.

El proyecto realizado en esta tesina reside en la importancia que tiene el protocolo HTTP y la interacción que hay con el servidor de Oracle. El manejo de información es importante hoy en día así como su seguridad e integridad por lo que debemos de saber manejar este tipo de herramientas de sofisticada calidad para así ofrecer un mejor servicio en todas las operaciones realizadas en la WEB.

Es importante señalar que el buen desarrollo de las aplicaciones así como sus políticas ofrecerán un mejor desempeño en las soluciones ofrecidas, el hecho de que Oracle sea una plataforma eficiente no quiere decir que su funcionamiento y performance sea optimo ya que esta funcionalidad integra, dependerá mucho de la experiencia de la gente que desarrolle estas soluciones.

El presente documento fue desarrollado con la finalidad de corroboran los conocimientos obtenidos en el taller de Oracle, y tratar de que sea de ayuda para futuros proyectos o simple documentación.

En la medida de las necesidades pondremos en práctica toda la experiencia adquirida para ofrecer servicios y ayudar en la mejora de las soluciones Web.

Bibliografía.

Guía de aprendizaje Oracle 9i
Editorial Osborne Mc Graw-Hill
Mexico, 2001

Schwarte, Joachim (Dr.)
El gran libro de HTML.Cómo publicar en Internet
Marcombo

Búsquedas en Internet

www.orafaq.com/
di.unipi.it/~ghelli/bdl/A97329_03/web.902/a92173/toc.htm
httpd.apache.org/
chebucto.ns.ca/~rakerman/oracle-port-table.html