Práctica 4: Servicio web

Objetivo

El objetivo de esta práctica es la instalación y configuración del servidor web Apache. Durante estos procesos se prestará especial atención al tema de la seguridad. También se restringirá el acceso a determinadas páginas ubicadas en el servidor según distintos criterios (IP, contraseña) y se realizarán una serie de pruebas para verificar tales restricciones.

1. Características de Apache

El proyecto servidor HTTP Apache es un esfuerzo para desarrollar y mantener un servidor web de código abierto para sistemas operativos modernos, entre los que se incluyen UNIX y Windows NT. El objetivo de este proyecto es proporcionar un servidor web seguro, robusto, eficiente y extensible que provea servicio HTTP actualizado al estándar HTTP del momento.

Desde 1996 es el servidor web más utilizado. En la actualidad, se estima que más del 60% de los sitios web emplean Apache.

Entre sus características destacan:

- Implementa el protocolo HTTP/1.1 (RFC2616)
- Multiplataforma: puede ejecutarse sobre Windows NT/9x, Netware 5.x y superiores, OS/2, y la mayoría de versiones de Unix.
- Modular: puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona y con la API de programación de módulos para el desarrollo de módulos específicos.
- Es extensible mediante módulos desarrollados por terceros. Como ejemplos pueden citarse *mod_php4* que implementa el lenguaje de programación del lado del servidor PHP y *mod_ssl* que dota a Apache de soporte SSL (*Secure Socket Layer v2/v3*) y TLS (*Transport Layer Security v1*) para seguridad y cifrado de las comunicaciones.
- Código abierto
- Activamente desarrollado y mantenido
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- La versión 2.0 puede ejecutarse en un modo híbrido multiproceso/multithread (sistemas con soporte *POSIX*).

2. Protocolo HTTP

El Protocolo de Transferencia de Hipertexto (*Hypertext Transfer Protocol*) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes web (navegadores) y los servidores HTTP. La especificación completa del protocolo HTTP/1.0

está recogida en el RFC 1945.

En el funcionamiento básico, el cliente establece una conexión TCP con el servidor, realiza una petición, el servidor le responde y se cierra la conexión. En la primera versión del protocolo 1.0, el cliente solo podía invocar tres operaciones en el servidor: GET para pedir una página, HEAD para pedir la cabecera de una página y POST para enviar datos a una URL.

Los problemas principales que existen en la versión 1.0 son de rendimiento. Estos problemas se documentan y analizan en `http://www.w3.org`. A continuación se destacan las conclusiones principales del citado análisis:

- El proceso de establecimiento de las conexiones del protocolo TCP es lento (conexión en tres pasos y ajuste de ventanas de recepción de datos). Además, como debe establecerse una conexión por cada elemento que hay en la página la transmisión de datos se ralentiza.
- Una conexión para transmitir 1 Kbyte de datos tarda alrededor de 500 ms.
- Tras cerrar una conexión TCP, el puerto del servidor utilizado en dicha conexión, se queda en estado TIME_WAIT un tiempo recomendado de 240 segundos, por lo que un servidor que reciba muchas peticiones puede agotar todos los puertos TCP (recordar, son 65535) y dejar al servidor sin posibilidad de enviar ningún tipo de dato. Esto supone un problema de escalabilidad muy importante.

A partir de este momento nos centraremos únicamente en el protocolo HTTP/1.1, cuya descripción está recogida dentro del RFC 2616. Sus principales características son:

- Conexiones persistentes: ya no se cierra la conexión tras el envío de cada parte de un documento, evitando la sobrecarga del establecimiento de conexiones TCP.
- Varias peticiones simultáneas: un cliente puede realizar varias peticiones utilizando una única conexión, sin esperar a la respuesta del servidor para cada una de ellas.
- Negociación del contenido: se asignan diferentes valores a las características de la comunicación.
- Nuevos métodos: junto a GET, POST y HEAD aparecen los métodos DELETE para borrar un recurso del servidor asociado al URI de borrado, TRACE para ver qué está recibiendo el servidor de lo que envía el cliente, PUT para enviar datos a un recurso asociado a una URI, PATCH para aplicar correcciones en un recurso asociado a una URI, COPY para copiar unos recursos identificados por una URI en otro lugar determinada URI en uno destino determinado, MOVE para mover el recurso identificado por la URI a otro lugar, DELETE para borrar un recurso asociado a una URI, LINK para establecer enlaces entre diferentes recursos, UNLINK para eliminar enlaces establecidos previamente por LINK, OPTIONS para que el cliente pueda obtener del servidor sus características, WRAPPED que permite unir varias peticiones y recubrirlas con algún tipo de filtrado (cifrado por ejemplo).
- Nuevo método de autentificación: en el RFC 2617 se describe un nuevo método de autentificación, en el cual las claves de acceso van cifradas por la red, al contrario de lo que ocurre en HTTP 1.0.

Esta versión HTTP 1.1 es un puente hacia un nuevo protocolo de transferencia de hipertexto: HTTP-NG (HTTP Next Generation), que pretende implementar nuevas funcionalidades, entre la que destaca el comercio electrónico. Sus criterios de diseño han sido:

- Simplicidad: no se debe abandonar el criterio introducido en HTTP 1.0: las cosas habituales deben ser sencillas, de forma que sea fácil implementar el protocolo.
- Rendimiento: debe ser eficiente trasmitiendo objetos en redes de comunicaciones.
- Asíncrono: las peticiones desde los clientes han de poderse hacer en paralelo a través de una única conexión.
- Seguridad: los objetos que se transmiten deben estar cifrados, sin forzar ninguna política de seguridad en particular.
- Autentificación: se debe poder autentificar a las dos partes de la conexión, así como a cualquier intermediario.
- Pagos en línea: el protocolo debe soportar la realización de pagos en línea.
- Servidores intermediarios: se debe soportar la comunicación entre servidores para el mantenimiento de caches, espejos de datos e intermediarios de comunicación (proxys).
- Visualización obligatoria: se debe poder obligar al cliente a mostrar ciertos datos acerca del objeto que se transmite, como el autor del objeto, el Copyright y la licencia.
- Información de registro: la información de registro (logs) ha de poder ser enviada entre diferentes servidores.
- Requerimientos de red: el protocolo debe trabajar de forma independiente de la capa de transporte de la que disponga, aunque debe funcionar especialmente bien con TCP, al ser el protocolo más utilizado en Internet.

3. Enunciado

El primer paso es la instalación de Apache. El paquete de instalación puede descargarse a partir de las siguientes direcciones:

`http://www.apache.org/` `http://apache.arrakis.es/`

Las instrucciones para la compilación e instalación de la versión 1.3 se adjuntan como documentación. Esta documentación junto con otra de utilidad puede encontrarse en:

`http://httpd.apache.org/docs/` `http://apache.arrakis.es/docs/`

Durante el proceso de instalación seguir en la medida de lo posible las directrices que aparecen en la documentación adjunta (*Guía para asegurar servidores web*, Apéndice A: Asegurando el servidor web Apache).

Repasa los módulos que se instalan por defecto y revisa si es necesario alguno adicional (piensa en la siguiente práctica). Documenta claramente los módulos instalados. Aunque aunque no es obligatorio, se valorará que el servicio HTTPS (módulo *mod_ssl*, que necesita la instalación previa en el sistema de las bibliotecas OpenSSL -ver referencias al final del guión-).

El siguiente paso es configurar el servidor. En este proceso sigue de nuevo las directrices

que aparecen en la documentación adjunta (*Guía para asegurar servidores web*, Apéndice A: Asegurando el servidor web Apache).

4. Pruebas

Una vez verificado el funcionamiento básico de Apache, se realizarán varias pruebas de control de acceso a los contenidos de un servidor web. Para ello, previamente debéis crear un sencillo sitio web con cinco páginas, donde la principal contendrá enlaces a las otras cuatro.

Debéis modificar los ficheros de configuración de Apache para que vuestro sitio web presente el siguiente comportamiento:

- La página principal tiene que ser disponible a todo el mundo.
- El primer enlace será accesible sólo desde una máquina del laboratorio
- El segundo enlace será accesible sólo desde las máquinas de la subred del laboratorio.
- El tercer enlace será de dominio público exceptuando a una máquina del laboratorio.
- El cuarto enlace será accesible tras autentificación mediante usuario y contraseña.

Se deja abierta la posibilidad de implementar restricciones de acceso adicionales.

5. Guión de la práctica

El guión de la práctica, breve pero completo, deberá especificar:

- **Entorno hardware**: características de la máquina de prueba.
- **Entorno software**: sistema operativo, versión, ...
- **Servidor web**: versión, justificación.
- **Instalación**: proceso, módulos instalados y eliminados ...
- **Configuración**: modificaciones/añadidos realizados sobre los ficheros de configuración (explicar brevemente mediante comentarios en los propios ficheros).
- **Control de acceso:** esquema utilizado, ficheros añadidos/modificados ...

6. Material auxiliar, referencias ...

- Sitio web del proyecto Apache: `http://httpd.apache.org`
- RFCs 1945, 2616
- *Guidelines on Securing Public Web Servers*:
`http://csrc.nist.gov/publications/nistpubs/800-44/sp800-44.pdf`
- *OpenSSL: the Open Source Toolkit for SSL/TLS:* `http://www.openssl.org`
- *mod_SSL: the Apache interface to OpenSSL:* `http://www.modssl.org`
- *Authentication, Authorization, and Access Control*:
`http://httpd.apache.org/docs/howto/auth.html`
- P. Laurie, B. Laurie, *Apache*, O'Reilly, 1999
- M.J. Kabir, *Apache Server Bible*, IDG Books, 1998