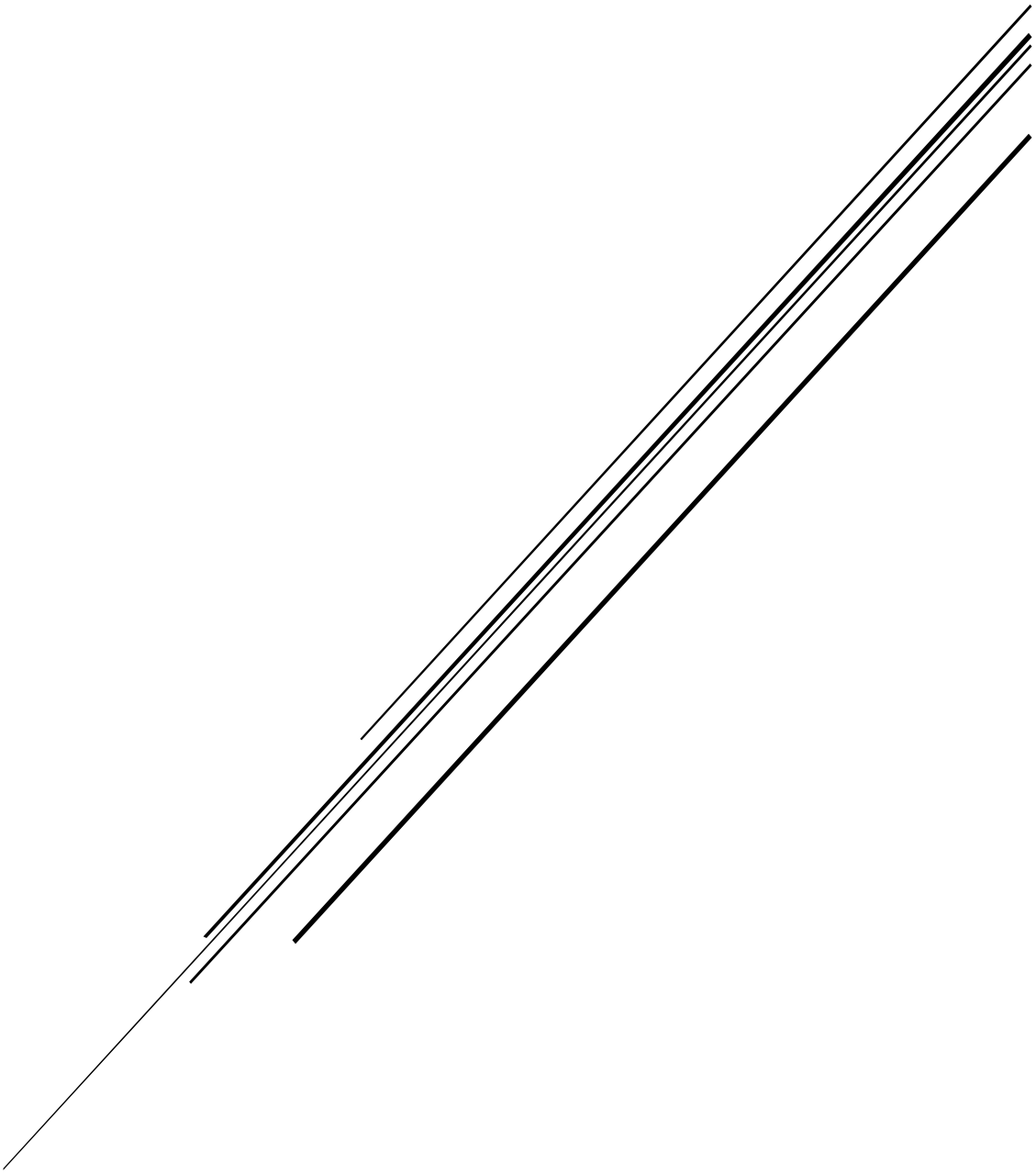
INVESTIGACIÓN SERVIDOR WEB

Joel Martínez Expósito



DAW2

Índice

Parámetros de administración de un servidor web:

Los parámetros de administración de un servidor web son aquellos que se pueden configurar dentro de un servidor web, algunos ejemplos podrían ser:

- ***Puertos de escucha***: Puerto en el que el servidor web escucha solicitudes
- ***DocumentRoot***: Ruta de los archivos donde se encuentra el servidor web
- ***ServerName***: Nombre del servidor web
- ***Timeouts****:* Tiempo que el servidor tardará en cerrar una conexión inactiva

Los parámetros de configuración de un servidor web se pueden modificar directamente en los archivos de configuración que controlan el comportamiento del servidor. Por ejemplo, en el servidor de apache se pueden modificar a través del archivo de configuración principal, en este caso httpd.conf.

Posibles módulos y sus funcionalidades:

Los módulos son programas de servicio que se pueden enlazar y cargar dinámicamente para ampliar la naturaleza de HTTP Server. Los módulos proporcionan una forma de ampliar la funcionalidad de los servidores web.

Algunos módulos saque puede tener un servidor web son:

mod_ssl: Habilita el cifrado para proporcionar el modelo HTTPS, asegurando que las comunicaciones entre el servidor y los clientes estén cifradas.

mod_rewrite: Permite la reescritura de las URL, lo que es útil para la creación de URLs amigables o para la manipulación avanzada de rutas.

mod_cache: Proporciona funcionalidades de almacenamiento en caché para mejorar el rendimiento al reducir la necesidad de servir contenido dinámico repetidamente.

mod_security: Añade una capa de seguridad que protege contra vulnerabilidades comunes como inyecciones sql y ataques de fuerza bruta.

mod_proxy: Facilita la creación de proxies inversos, balanceadores de carga y servidores caché, lo que permite manejar varias aplicaciones detrás de un solo servidor web.

mod_auth_basic: Implementa autenticación básica HTTP, solicitando a los usuarios que proporcionen un nombre de usuario y contraseña para acceder a determinadas áreas.

Virtual Hosts

En este punto se explicará la definición de virtual host, definiendo sus ventajas y sus utilidades.

Un hosting virtual, host, o anfitrión virtual, es una técnica que **permite a un servidor web alojar múltiples sitios web en la misma máquina física**. Esto se logra mediante la asignación de nombres de dominio o direcciones IP específicas a cada sitio web, lo que permite al servidor identificar y enrutar las solicitudes de manera adecuada.

En otras palabras, los virtual hosts permiten que un solo servidor web funcione como si fuera varios servidores independientes.

Con virtual hosts, al alojar varios sitios en un servidor compartido, se maximiza la utilización de recursos, como el espacio de almacenamiento y la potencia de procesamiento. Además, al reducir los servidores físicos individuales, se reducen los gastos en hardware, energía y mantenimiento. Gracias a su simplicidad es fácil agregar nuevos sitios web sin la necesidad de adquirir hardware adicional. Por último, otra ventaja de utilizar virtual hosts es que cada sitio tiene su propia configuración, como software de servidor web y base de datos, lo que permite una personalización completa.

Existen varios tipos de virtual hosts, cada uno con sus propias características y casos de uso. Los principales tipos incluyen los siguientes:

Virtual hosts basados en nombre (Name-Based Virtual Hosts)

En este enfoque, se utiliza el nombre de dominio en el encabezado HTTP de la solicitud para enrutar la petición al sitio web correspondiente. Es el método más común y eficiente para alojar múltiples sitios en una sola dirección IP.

Virtual hosts basados en dirección IP (IP-Based Virtual Hosts)

Aquí, se asigna una dirección IP única a cada sitio web alojado en el servidor. Cada dirección IP se asocia con un sitio específico. Aunque menos común, es útil cuando

se necesita garantizar la compatibilidad con navegadores antiguos o sistemas que no admiten el enrutamiento basado en nombre.

Virtual hosts basados en puertos (Port-Based Virtual Hosts)

Este enfoque implica que cada sitio web se asocie con un puerto específico en la dirección IP del servidor. Es útil cuando se desean múltiples sitios web en una única dirección IP, pero sin utilizar el enrutamiento basado en nombre.

Virtual hosts con SSL/TLS (Secure Virtual Hosts)

Estos virtual hosts están diseñados para sitios web que requieren conexiones seguras a través de HTTPS. Permiten la configuración de certificados SSL/TLS individuales para cada sitio alojado, garantizando la seguridad de la comunicación.

En cuanto a la **configuración de virtual hosts**, varía el servidor web que esté utilizando. A continuación se explicara como configurarlo en el servidor web Nginx.

Nginx utiliza archivos de configuración ubicados en el directorio “sites-available” para definir virtual hosts. Cada virtual host se crea como un bloque de servidor y se especifica la dirección IP y el puerto. No obstante, es importante consultar la documentación específica de su servidor web para obtener instrucciones detalladas sobre la configuración de servidores virtuales en cada caso.

MECANISMOS DE AUTENTICACIÓN Y DE ACCESO AL SERVIDOR

Los mecanismos de autenticación de un servidor son métodos que validan la identidad del usuario antes de permitir el acceso al servidor. Su objetivo principal es garantizar que únicamente usuarios autorizados puedan interactuar con los recursos del servidor, protegiendo así la confidencialidad y la integridad de los datos. Algunos de los mecanismos de autenticación más comunes incluyen:

Autenticación Basada en Contraseña:

El servidor verifica que la contraseña proporcionada por el usuario coincida con la que tiene almacenada. Este es uno de los métodos más sencillos y extendidos, aunque su efectividad depende de la complejidad de la contraseña.

Autenticación Basada en Tokens

Se genera un token de un solo uso, que es una cadena alfanumérica o un número temporal (generalmente de uso temporal). Si el token proporcionado por el usuario coincide con el generado por el servidor, el acceso es concedido. Un ejemplo común de este tipo de autenticación es el uso de tokens de autenticación en dos factores (2FA), donde un código temporal se envía al dispositivo del usuario y debe ser ingresado junto con la contraseña.

Autenticación Basada en IP0

El servidor compara la dirección IP desde la que el usuario intenta acceder con una lista de direcciones IP autorizadas. Si la IP coincide con una de las almacenadas, se otorga el acceso. Este método es útil en entornos donde los usuarios acceden desde ubicaciones conocidas y de confianza, pero puede ser vulnerable a cambios en la IP, como en redes dinámicas o al utilizar VPNs.

Estos mecanismos permiten garantizar que solo personas autorizadas puedan acceder al servidor, lo que protege los datos y recursos de posibles accesos no deseados. Además, es común que se combinen varios de estos métodos, como el uso de contraseñas y autenticación en dos factores, para mejorar la seguridad global del sistema.

CERTIFICADOS DIGITALES

Un certificado digital en un servidor web es un archivo electrónico que vincula la identidad de un sitio web con una clave pública, lo que permite establecer una conexión segura entre el servidor y los clientes (usuarios o navegadores web). Es emitido por una Autoridad Certificadora (CA) confiable, que verifica la identidad del propietario del dominio o servidor.

¿Para qué sirve un certificado digital en un servidor web?

Un **certificado digital** en un servidor web es un archivo electrónico que vincula la identidad de un sitio web con una clave pública, lo que permite establecer una conexión segura entre el servidor y los clientes (usuarios o navegadores web). Es emitido por una **Autoridad Certificadora (CA)** confiable, que verifica la identidad del propietario del dominio o servidor.

Los navegadores web utilizan el certificado digital para verificar que el servidor web es legítimo y que el usuario se está conectando al sitio web correcto, y no a un sitio fraudulento. Esto previene ataques como el **phishing** y el **man-in-the-middle**.

Cuando un sitio web tiene un certificado digital válido, los navegadores muestran un icono de **candado** junto a la URL y utilizan el prefijo "https://" en lugar de "http://". Esto indica que el sitio es seguro, lo que genera confianza en los usuarios para interactuar y compartir información con el sitio web.

Existen varios tipos de certificados digitales, cada uno con su utilidad y sus ventajas;

Certificado de validación de dominio (D - Domain Validation)

En este tipo de certificados la verificación es básica, la autoridad solo comprueba que el solicitante tiene control sobre el dominio. No se verifica la identidad del propietario del sitio web. Sus principales ventajas son su rápida emision, su bajo costo o gratuito y el cifrado básico.

Certificado de Validación de Organización (OV - Organization Validation)

Es parecido al anterior, pero este también verifica la existencia legal de la organizacion solicitante. Esto incluye revisar registros comerciales y documentos corporativos. Sus principales ventajas son la confianza que muestra a los visitantes que el sitio pertenece a una organización legítima además de que proporciona tanto cifrado de datos como autenticación del servidor

Certificado de Validación Extendida (EV - Extended Validation)

Este es el nivel más alto de verificación. La CA realiza un proceso exhaustivo para validar la identidad legal y física del solicitante, y verifica la propiedad del dominio. La principal ventaja que ofrece junto con las de los demás certificados es la protección contra el phishing, los ciberdelincuentes suelen evitar atacar sitios con certificados EV debido a la complejidad de obtener un certificado falso.

Certificados Wildcard (Certificados Comodín)

Estos certificados protegen un dominio principal y todos sus subdominios (por ejemplo, proteger tanto "example.com" como "subdomain.example.com"). Sus principales ventajas son que protege múltiples subdominios con un solo certificado, lo que simplifica la gestión de certificados. Es más económico que comprar certificados individuales para cada subdominio.

Pruebas de funcionamiento y rendimiento del servidor Web

1. Pruebas de carga (Load Testing)

Las pruebas de carga evalúan cómo responde el servidor bajo un número creciente de usuarios o solicitudes concurrentes. Miden el rendimiento en condiciones normales de operación, enfocándose en el tiempo de respuesta y uso de recursos, usando herramientas como JMeter y Locust.

2. Pruebas de estrés (Stress Testing)

Las pruebas de estrés someten al servidor a una carga extrema para identificar su límite de capacidad antes de fallar. Evalúan su capacidad de recuperación y revelan cuellos de botella, empleando herramientas como Gatling y Siege.

3. Pruebas de latencia (Latency Testing)

Estas pruebas miden el tiempo que tarda el servidor en procesar y responder a una solicitud, fundamental para garantizar una buena experiencia de usuario. Herramientas como Pingdom y GTmetrix son utilizadas para medir la latencia y optimizar tiempos de carga.

4. Pruebas de capacidad (Capacity Testing)

Determinan cuántas solicitudes o conexiones simultáneas puede manejar el servidor antes de que su rendimiento se degrade. Herramientas como Blazemeter permiten simular altos volúmenes de usuarios para evaluar la capacidad.

5. Pruebas bajo condiciones de red adversas

Simulan condiciones de red como alta latencia o pérdida de paquetes para verificar cómo maneja el servidor situaciones adversas, utilizando simuladores de redes como WANem y NetEm.

6. Pruebas de disponibilidad y tiempo de actividad (Uptime Testing)

Monitorean el tiempo de actividad del servidor para garantizar que esté siempre disponible, alertando sobre caídas del servicio. Uptime Robot y Site24x7 son herramientas comunes para este tipo de pruebas.

7. Pruebas de seguridad (Security Testing)

Evalúan la seguridad del servidor contra vulnerabilidades como ataques DoS o inyecciones SQL. Herramientas como OWASP ZAP y Nikto analizan configuraciones inseguras y posibles brechas.

8. Pruebas de funcionalidad (Functional Testing)

Garantizan que el servidor cumple con las funciones esperadas, como la correcta carga de contenido y ejecución de scripts. Herramientas como Selenium automatizan estas pruebas para verificar la interacción del usuario.

9. Monitoreo en tiempo real

Monitorean en tiempo real el uso de recursos del servidor (CPU, memoria, red) para detectar problemas de rendimiento. Herramientas como Nagios y Zabbix proporcionan estadísticas y alertas en tiempo real.

10. Pruebas de escalabilidad (Scalability Testing)

Evalúan cómo el servidor responde al aumentar gradualmente la carga, ya sea mediante escalado horizontal o vertical. K6 y Tsung son herramientas comunes para medir la capacidad de escalar del servidor.