1 PROTOCOLO HTTP Y HTTPS.

El protocolo HTTPS (protocolo basado en el protocolo HTTP, destinado a la transferencia segura de datos mediante cifrado, es decir, es la versión segura de HTTP) sirve para transferir la información de forma segura y conservar la información de forma confidencial. Al contrario que el protocolo HTTP (protocolo usado en cada transacción de la World Wide Web).

El protocolo HTTPS permite que la información viaje de forma segura entre el cliente y el servidor, al contrario que el protocolo HTTP que envía la información en texto claro, esto es, cualquiera que accediese a la información transferida entre el cliente y el servidor puede ver el contenido exacto y textual de la información.

Para asegurar la información, el protocolo HTTPS requiere de certificados. Siempre y cuando sean validados, la información será transferida cifrada. Pero cifrar la información requiere un tiempo de computación, por lo que será perjudicado el rendimiento del servidor web. No es necesaria que toda la información sea cifrada. Tal vez solamente es necesaria que sea cifrada la autenticación a dicha información.

Un servidor web, como Apache, puede emitir certificados, pero puede que en algún navegador s ea interpretado como peligroso, esto suele ser debido a que los navegadores poseen en su configuración una lista de Entidades Certificadoras que verifican, autentican y dan validez a los certificados. Los navegadores, solamente confían en quien confían. Eso no quiere decir que no puedas crear tus certificados en un servidor web, de hecho muchas empresas lo hacen, sobre todo para sitios internos o externos en los que solamente puede acceder personal autorizado por la propia empresa.

El protocolo HTTPS utiliza cifrado sobre SSL/TLS (protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet) que proporcionan autenticación y privacidad. Entonces, si necesitas que la información viaje cifrada debes emplear el protocolo HTTPS, en caso contrario el protocolo HTTP. Hay que dejar claro que la utilización del protocolo HTTPS no excluye ni impide el protocolo HTTP, los dos pueden convivir en un mismo dominio.

En el **protocolo HTTP** cuando escribes una dirección URL en el navegador, por ejemplo http://www.debian.org/index.es.html, antes de ver la página en el navegador existe todo un juego de protocolos, sin profundizar en todos ellos básicamente lo que ocurre es lo siguiente: se traduce el dominio DNS (sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada, por ejemplo: apache.org determina un dominio org (organización) y un subdominio que identifica en este caso la

máquina o conjunto de máquinas de nombre apache) por una IP, una vez obtenida la IP se busca en ella si un servidor web aloja la página solicitada en el puerto 80 (número utilizado en las comunicaciones cliente/servidor, en transmisiones TCP o UDP comprendido entre 1 y 65535, que indica por donde tiene lugar la conexión con un servidor. Están estandarizados, esto es, un servidor suele estar activo siempre por definición en un puerto determinado, pero éste puede que sea modificado en la configuración del servidor. Por ejemplo un servidor web espera en el puerto TCP 80), puerto TCP (es uno de los protocolos fundamentales en Internet. Garantiza que los datos serán entregados en su destino sin errores y una vez recogidos ponerlos en el mismo orden en que se transmitieron) asignado por defecto al protocolo HTTP. Si el servidor web aloja la página ésta será transferida a tu navegador.

Sin embargo cuando escribes en el navegador una dirección URL con llamada al **protocolo HTTPS**, el procedimiento es similar al anterior pero un poco más complejo, así se traduce el dominio DNS por una IP, con la IP se busca el servidor web que aloja la página solicitada en el puerto 443, puerto TCP asignado por defecto al protocolo HTTPS, pero ahora antes de transferir la página a tu navegador se inicia una negociación SSL, en la que entre otras cosas el servidor envía su certificado -el navegador aunque es poco habitual también puede enviar el suyo-. Si el certificado es firmado por un Entidad Certificadora de confianza se acepta el certificado y se cifra la comunicación con él, transfiriendo así la página web de forma cifrada.

Puedes hacer que un servidor web para una determinada página espere los protocolos HTTP y HTTPS en puertos TCP distintos del 80 y 443 respectivamente. Eso sí, cuando visites la página web en la dirección URL debes especificar el puerto TCP, por ejemplo: http://www.tupagina.local:8080, de esta forma el servidor web espera la petición de la página www.tupagina.local en el puerto 8080; del mismo modo en la dirección URL: https://www.tupagina.local:4333 espera la petición de la página en el puerto 4333.

Como ves, puedes configurar los puertos, pero ten en cuenta que cualquiera que quisiera acceder a esas páginas debería saber el puerto TCP de la solicitud. Entonces, quiere decir que aunque no escribas el puerto TCP en las direcciones URL estas se interpretan en el puerto 80 y 443 para el protocolo HTTP y HTTPS respectivamente.

Es lo mismo escribir http://www.tupagina.local:80 que http://www.tupagina.local y es lo mismo escribir https://www.tupagina.local:443 que https://www.tupagina.local

En http://www.warriorsofthe.net/index.html puedes encontrar un vídeo sobre el funcionamiento de Internet.

1.1 Características Técnicas

El sistema HTTPS utiliza un cifrado basado en SSL/TLS para crear un canal cifrado. De este modo se consigue que la información sensible (usuario y claves de paso normalmente) no pueda ser usada por un atacante que haya conseguido interceptar la transferencia de datos de la conexión, ya que lo único que obtendrá será un flujo de datos cifrados que le resultará imposible de descifrar.

El puerto estándar para este protocolo es el 443.

Diferencias con HTTP

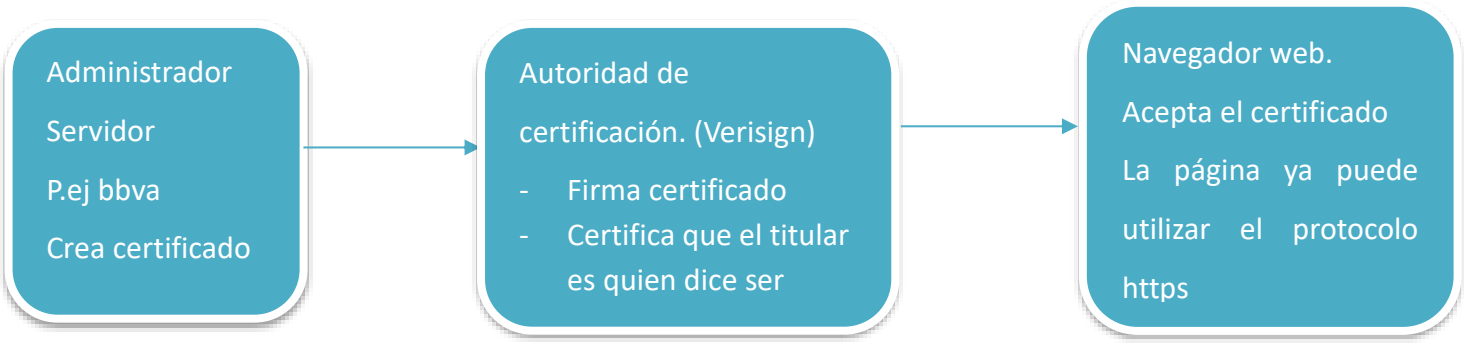
En el protocolo HTTP las URLs comienzan con "http://" y utilizan por defecto el puerto 80, Las URLs de HTTPS comienzan con "https://" y utilizan el puerto 443 por defecto.

HTTP es inseguro y está sujeto a ataques man-in-the-middle y eavesdropping que pueden permitir al atacante obtener acceso a cuentas de un sitio web e información confidencial. HTTPS está diseñado para resistir esos ataques y ser seguro.

Configuración del Servidor

Para preparar un servidor web que acepte conexiones HTTPS, el administrador debe crear un Certificado de clave pública para el servidor web.

Este certificado debe estar firmado por una Autoridad de certificación para que el navegador web lo acepte. La autoridad certifica que el titular del certificado es quien dice ser. (Los navegadores web generalmente son distribuidos con los certificados raíz firmados por la mayoría de las Autoridades de Certificación por lo que estos pueden verificar certificados firmados por ellos).



2 SERVIDORES Y APLICACIONES LIBRES Y PROPIETARIAS

Una plataforma web es el entorno de desarrollo de software empleado para diseñar y ejecutar un sitio web.

En términos generales, **una plataforma web consta de cuatro componentes básicos:**

- **El sistema operativo**, bajo el cual opera el equipo donde se hospedan las páginas web y que representa la base misma del funcionamiento del computador. En ocasiones limita la elección de otros componentes. (Linux, Windows, MAC)

- **El servidor web** es el software que maneja las peticiones desde equipos remotos a través de la Internet. En el caso de páginas estáticas, el servidor web simplemente provee el archivo solicitado, el cual se muestra en el navegador. En el caso de sitios dinámicos, el servidor web se encarga de pasar las solicitudes a otros programas que puedan gestionarlas adecuadamente. (Apache, IIS)

- **El gestor de bases de datos** se encarga de almacenar sistemáticamente un conjunto de registros de datos relacionados para ser usados posteriormente. (MySQL, SQLServer)

- **Un lenguaje de programación** interpretado que controla las aplicaciones de software que corren en el sitio web. (php, ASP, C#, phyton)

Diferentes combinaciones de los cuatro componentes señalados, basadas en las distintas opciones de software disponibles en el mercado, dan lugar a numerosas plataformas web.

La plataforma LAMP trabaja enteramente con componentes de software libre y no está sujeta a restricciones propietarias. El nombre LAMP surge de las iniciales de los componentes de software que la integran:

- Linux: Sistema operativo.
- Apache: Servidor web.
- MySQL: Gestor de bases de datos.
- PHP: Lenguaje interpretado PHP, aunque a veces se sustituye por Perl o Python.

La plataforma WISA está basada en tecnologías desarrolladas por la compañía Microsoft; se trata, por lo tanto, de software propietario. La componen los siguientes elementos:

- Windows: Sistema operativo.
- Internet Information Services: servidor web.
- SQL Server: gestor de bases de datos.
- ASP o ASP.NET: como lenguaje para scripting del lado del servidor.

Existen otras plataformas, como por ejemplo la configuración Windows-Apache-MySQL-PHP que se conoce como **WAMP**. Es bastante común pero sólo como plataforma de desarrollo local.

De forma similar, un servidor Windows puede correr con MySQL y PHP. A esta configuración se la conoce como **plataforma WIMP**.

Plataforma XAMPP paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. El nombre es en realidad un acrónimo: X, Apache, MariaDB/MySQL, PHP, Perl.

Existen muchas otras plataformas que trabajan con distintos sistemas operativos (Unix, MacOS, Solaris), servidores web (incluyendo algunos que se han cobrado relativa popularidad como Lighttpd yLiteSpeed), bases de datos (Postgre SQL) y lenguajes de programación.