📁 LPIC-2 / 🌍 Examen 208 - Servicios Web

208.4 Implementar un servidor web seguro

Teoría

La información transmitida a través de HTTP viaja en texto plano y puede ser interceptada y leída. Implementar un servidor web seguro significa cifrar esta comunicación para garantizar la confidencialidad, la integridad y la autenticación.

HTTPS (HTTP sobre TLS/SSL):

- **Propósito:** Proporciona una conexión segura entre el cliente y el servidor cifrando el tráfico HTTP. La URL comienza con `https://` y el puerto por defecto es 443.
- **Tecnología Subyacente:** TLS (Transport Layer Security), anteriormente conocido como SSL (Secure Sockets Layer).
- **Cómo Funciona (Resumen):**
    1. El cliente se conecta al puerto 443 del servidor.
    2. Se inicia un "apretón de manos" (handshake) TLS: cliente y servidor negocian la versión de TLS y los algoritmos de cifrado a usar.
    3. El servidor envía su **certificado TLS/SSL** al cliente.
    4. El cliente verifica la validez del certificado (si fue emitido por una Autoridad de Certificación (CA) de confianza, si el nombre de dominio coincide, si no ha caducado/revocado).
    5. Si la verificación es exitosa, el cliente y el servidor utilizan las claves de sesión negociadas para cifrar y descifrar toda la comunicación HTTP subsiguiente.

Certificados TLS/SSL:

- **Propósito:** Un certificado digital vincula una **clave pública** a una **identidad** (un nombre de dominio, una organización). Es emitido y firmado por una **Autoridad de Certificación (CA)**. Los navegadores confían en los certificados firmados por CAs reconocidas.
- **Clave Privada:** Un archivo secreto que reside solo en el servidor. Se utiliza para descifrar los datos cifrados por la clave pública correspondiente y para firmar digitalmente datos durante el handshake TLS. **Debe mantenerse muy segura.**
- **Clave Pública:** Se incluye en el certificado. Se utiliza para cifrar datos que solo la clave privada correspondiente puede descifrar y para verificar firmas digitales creadas con la clave privada.
- **Autoridad de Certificación (CA):** Una entidad de confianza que verifica la identidad del solicitante y firma su certificado. Los sistemas operativos y navegadores vienen con una lista de CAs raíz preinstaladas en las que confían.
- **Tipos de Certificados:**
    - **Autofirmado (Self-Signed):** Firmado por la propia entidad que lo creó, no por una CA de confianza. Son útiles para pruebas internas o entornos de desarrollo, pero los

navegadores mostrarán una advertencia de seguridad porque no confían en la CA (que eres tú mismo).
- **Emitido por CA de Confianza:** Firmado por una CA reconocida (ej: Let's Encrypt - gratis, Sectigo, DigiCert). Son confiados por navegadores y sistemas operativos.

Proceso de Obtención de un Certificado (Emitido por CA):

1. **Generar Clave Privada:** En el servidor, crea una clave privada (ej: usando `openssl genrsa`).
2. **Crear Solicitud de Firma de Certificado (CSR - Certificate Signing Request):** Crea un archivo CSR que contiene tu clave pública e información sobre tu identidad (nombre de dominio común, organización, etc.) (ej: usando `openssl req -new`).
3. **Enviar el CSR a una CA:** Envía el archivo CSR a una CA. La CA verificará tu identidad (el nivel de verificación depende del tipo de certificado).
4. **Recibir el Certificado Firmado:** La CA te devuelve el certificado público firmado (ej: archivos `.crt`, `.cer`, `.pem`).
5. **Obtener Certificados Intermedios/Cadena:** A menudo, la CA raíz no firma directamente tu certificado. Hay una cadena de certificados: Tu Certificado -> CA Intermedia -> CA Raíz. Necesitas descargar los certificados intermedios de la CA y a menudo combinarlos en un archivo de "cadena" para que los navegadores puedan verificar la confianza hasta la raíz.
6. **Instalar Clave, Certificado y Cadena en el Servidor Web:** Coloca estos archivos en un lugar seguro en el servidor (ej: `/etc/ssl/certs/`, `/etc/ssl/private/`) y configura el servidor web para usarlos.

Configuración del Servidor Web para HTTPS:

- **Módulo SSL/TLS:** Los servidores web necesitan un módulo para manejar TLS/SSL. Apache usa `mod_ssl` (a menudo instalado por defecto o como paquete separado). Nginx tiene el soporte SSL/TLS integrado.
- **Configuración en Virtual Host / Server Block:** Se configura una entrada para el puerto 443.
    - **Apache:** En un bloque `<VirtualHost *:443>`. Directivas clave:
        - `SSLEngine On`: Habilita SSL/TLS para este Virtual Host.
        - `SSLCertificateFile <ruta_a_certificado_publico>`: Ruta al archivo del certificado público del servidor (`.crt`, `.pem`).
        - `SSLCertificateKeyFile <ruta_a_clave_privada>`: Ruta al archivo de la clave privada del servidor (`.key`, `.pem`).
        - `SSLCACertificateFile <ruta_a_archivo_cadena>`: Ruta al archivo que contiene los certificados intermedios/raíz (cadena de confianza).
        - **Ubicación de Configuración (Diferencias):** En Debian, suele estar en un archivo `.conf` en `/etc/apache2/sites-available/` y habilitado en `sites-enabled/`, o en archivos de configuración SSL incluidos desde `/etc/apache2/conf-available/` y habilitados en `conf-`

`enabled/` (usando `a2enconf ssl`). En Red Hat, se configura dentro del bloque `VirtualHost` en archivos `.conf` en `/etc/httpd/conf.d/` o `/etc/httpd/conf.d/ssl.conf`.

- **Nginx:** En un bloque `server` que escucha en el puerto 443 (`listen 443 ssl;`). Directivas clave:
  - `ssl_certificate <ruta_a_certificado_publico>;`: Ruta al archivo del certificado público.
  - `ssl_certificate_key <ruta_a_clave_privada>;`: Ruta al archivo de la clave privada.
  - `ssl_trusted_certificate <ruta_a_archivo_cadena>;`: Ruta al archivo de cadena (puede ser necesario concatenar el certificado público y la cadena).
  - **Ubicación de Configuración:** Se configura directamente dentro del bloque `server` o en un archivo incluido desde allí.
- **Redirección HTTP a HTTPS:** Es común redirigir automáticamente a los usuarios que intentan acceder por HTTP (puerto 80) a la versión segura HTTPS (puerto 443). Esto se configura en el Virtual Host/Server Block del puerto 80.
- **Firewall:** Asegúrate de que el firewall permite el tráfico TCP entrante al puerto 443.

Generación de Certificado Autofirmado (para pruebas):

Puedes generar una clave privada y un certificado autofirmado en un solo paso usando `openssl`.`openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt` Te pedirá información (país, ciudad, nombre común - ¡debe coincidir con el nombre de host que usarás para acceder!).

Prueba de HTTPS:

- Acceder a `https://<ip_servidor>` o `https://<nombre_host_servidor>` en un navegador. Si usas un certificado autofirmado, verás una advertencia de seguridad.
- `curl -v https://<ip_servidor>`: La opción `-v` muestra detalles del proceso SSL/TLS y la verificación del certificado. Usa `-k` (`curl -vk https://...`) para ignorar errores de certificado (útil con autofirmados).
- `openssl s_client -connect <ip_servidor>:443`: Una herramienta de bajo nivel para depurar conexiones TLS/SSL.