Tecnologías y herramientas para el desarrollo web (2)

Seguridad entre cliente-servidor

¿Todas las páginas web que están alojadas en un sitio deben ser accesibles por cualquier
usuario?
¿Todas las accesibles deben enviar la información sin cifrar, en texto plano?
¿Es necesario que todo el trasiego de información navegador-servidor viaje cifrado?

Niveles de Comunicación Cliente/Servidor Seguros
El proceso de identificación del usuario constituye una parte fundamental en la creación de un sistema seguro.

• La forma más sencilla de seguridad es la ***autenticación*** que simplemente identifica al
cliente. Constituye la primera puerta que tiene que pasar un usuario a través de un sistema seguro. El procedimiento de entrada o conexión sólo comprueba el nombre de usuario y contraseña.

• El siguiente nivel de seguridad, la ***autorización***, permite o deniega acceso a los servicios del sistema.

• La ***certificación*** ofrece mayor nivel de seguridad con respecto a la autenticación. Requiere una tercera parte de confianza que permita comprobar tanto la identidad del servidor como del cliente.

Existe la posibilidad de asegurar la información sensible que viaja entre el navegador y el servidor, pero esto repercutirá en un mayor consumo de recursos del servidor, puesto que asegurar la información implica en que ésta debe ser cifrada, lo que significa computación algorítmica.

El cifrado al que nos referimos es el cifrado de clave pública o asimétrico: clave pública (en adelante kpub) y clave privada (en adelante kpriv).

La kpub interesa publicarla para que llegue a ser conocida por cualquiera, la kpriv no interesa que nadie la posea, solo el propietario de la misma.

Ambas son necesarias para que la comunicación sea posible, una sin la otra no tiene sentido, así una información cifrada mediante la kpub solamente puede ser descifrada mediante la kpriv y una información cifrada mediante la kpriv solo puede ser descifrada mediante la kpub.

En el cifrado asimétrico podemos estar hablando de individuos o de máquinas, en nuestro caso hablamos de máquinas y de flujo de información entre el navegador (A) y el servidor web (B).

Por lo tanto, el cifrado asimétrico utiliza dos claves, una de cifrado (pública) y otro de descifrado (privada). A menudo un servidor utiliza este algoritmo para el proceso de cifrado o simplemente para realizar transacciones.

A diferencia del cifrado asimétrico, el cifrado de clave simétrica utiliza la misma clave para cifrar y descifrar. Los dos hosts deben mantener la clave en secreto.

Algoritmos más utilizados

- Clave pública o asimétrica:
  - RSA (Rivest, Shamir y Adlemn)
  - DSA (Digital Signature Algorithm)
  - Diffie--Hellman

- Clave simétrica:
  - DES (Data Encryption Standard)
  - 3DES
  - AES (Advanced Encryp<on Standard)
  - RC2/RC4 (Rivest Ciphers)

Seguridad a nivel de Sockets (SSL)

SSL (Socket Secure Layer) Es un protocolo que nos proporciona una canal (socket) seguro entre un cliente y un servidor.

Tiene básicamente **tres características:**

- Es una capa transparente a la aplicación
- Permite comprobar la autenticidad del servidor (y opcionalmente del cliente).
- Todos los datos circulan encriptados.

SSL se puede aplicar a cualquiera de los protocolos de aplicación utilizando alguno de los siguientes **enfoques**:

- Puertos separados
  - Se eligen dos puertos separados, uno para las conexiones seguras y el otro para las no seguras, p.e. HTTP en el 80 y HTTPS en el 443

- Negociación de la conexión
  - Esta opción consiste en que un servidor instalado en el mismo puerto pueda negociar entre utilizar el protocolo tradicional sin encriptación, o encriptarlo con SSL, p.e. SMTP y POP3

Origen SSL/TLS

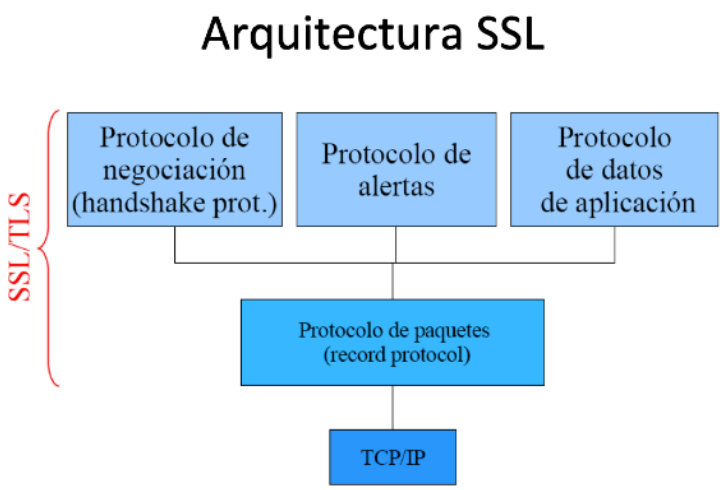
- SSL desarrollado por Netscape (v2, v3)
- Estandarizado por IETF como TLS (Transport Layer Secure)
- TLS 1.0 sería SSL 3.1 (parecido a SSL 3.0 pero incompatible)
- PCT y STPL propuestas de Microsoft no aceptadas.
- Versión actual TLS 1.1 (RFC 4346)
- Comunicación segura a través de Internet
  - Confidencialidad
  - Autenticación (del servidor)
  - Integridad
- Corre debajo de los protocolos usuales
  - HTTP, FTP, POP...

Componentes

- **Encriptación simétrica**
- **Claves públicas para**
  - **Autenticación**
  - **Intercambio de claves**
- PKI centralizado
  - Autoridades de certificación

  - Certificados X.509
- Algoritmos negociados
  - Claves públicas: RSA, DSA, Diffie--Hellman
  - Cifrados simétricos: RC2, RC4, IDEA, DES, 3DES, AES
- Funciones de hash: MD5, SHA--1

Arquitectura SSL



Funcionamiento SSL

- **Handshake**
  - Se autentica el servidor, y se realiza un key agreement entre cliente y servidor (se comprueba que ambos están de acuerdo con la comunicación)

El handshake tiene tres **propósitos:**

- Acordar los algoritmos criptográficos a usar durante la conexión.
- Acordar las claves criptográficas de sesión.
- Identificar al servidor (y opcionalmente al cliente)

1- El cliente envía al servidor una lista con los algoritmos criptográficos que soporta, junto con un número aleatorio, que usará para el key agreement.

2- El servidor elige un algoritmo criptográfico de la lista y se lo envía al cliente junto con su certificado digital (opcional) y su número aleatorio que había enviado al cliente firmado con su clave privada.

3- El cliente comprueba la validez del certificado, extrae la clave pública y comprueba que el número haya sido firmado correctamente.

- **Transmisión de datos segura**
  - Los datos son transmitidos de forma segura.

- **Cierre de la conexión**
  - Cierre seguro