

Índice

- Protocolos Seguros para el Web..... 1

Por Daniel Sepulveda

Protocolos Seguros para el Web

Se discuten SSH, SSL, TSL y HTTPS, los protocolos utilizados en la actualidad para intercambiar información de manera de hacer difícil que esta sea interceptada por terceros. Contar con protocolos seguros es importante tanto por las preocupaciones relacionadas con la privacidad como para permitir el comercio electrónico.

Seguridad en la Web

Dado el gran auge que hoy en día tiene Internet, su uso se ha masificado enormemente. Desde páginas meramente informativas hasta sitios interactivos usando tecnologías nuevas.

Empresas de diversa índole ya usan la Internet para comunicarse y el problema principal que surgió es la confiabilidad en que lo que se esta comunicando no sea visto por personas que puedan hacer mal uso de dicha información.

Por ejemplo, las tiendas comerciales ya están dando la posibilidad de realizar compras por la Web, pero el principal talón de Aquiles lo constituye la inseguridad que causa dar un número de tarjeta de crédito para pagar la compra.

O cosas tan simples como cuando uno envía un mail y no querer que nadie lo lea sino el destinatario.

A raíz de todo esto surgieron tecnologías que persiguen mejorar la seguridad de todas estas comunicaciones.

Seguridad en la transmisión

La seguridad de este tipo se basa en el hecho de poder encriptar los mensajes que se envían por a red entre un servidor y un cliente y que solo ellos puedan descifrar los contenidos a partir de una clave común conocida solo por los dos.

Para llevar a cabo esta seguridad se crearon diversos protocolos basados en esta idea:

- SSH: Usado exclusivamente en reemplazo de telnet
- SSL: Usado principalmente en comunicaciones de hipertexto pero con posibilidad de uso en otros protocolos
- TSL: Es del mismo estilo del anterior.
- HTTPS: Usado exclusivamente para comunicaciones de hipertexto

SSH (Secure Shell)

Este protocolo fue diseñado para dar seguridad al acceso a computadores en forma remota.

Cumple la misma función que telnet o rlogin pero además, usando criptografía, logra seguridad con los datos.

A diferencia de telnet u otro servicio similar, SSH utiliza el puerto 22 para la comunicación y la forma de efectuar su trabajo es muy similar al efectuado por SSL.

Para su uso se requiere que por parte del servidor exista un demonio que mantenga continuamente en el puerto 22 el servicio de comunicación segura, el sshd.

El cliente debe ser un software tipo TeraTerm o Putty que permita la hacer pedidos a este puerto 22 de forma cifrada.

La forma en que se entabla una comunicación es en base la misma para todos los protocolos seguros:

- El cliente envía una señal al servidor pidiéndole comunicación por el puerto 22.
- El servidor acepta la comunicación en el caso de poder mantenerla bajo encriptación mediante un algoritmo definido y le envía la llave publica al cliente para que pueda descifrar los mensajes.
- El cliente recibe la llave teniendo la posibilidad de guardar la llave para futuras comunicaciones o destruirla después de la sesión actual.

Se recomienda que si se esta en un computador propio, la clave sea guardada, en otro caso, destruirla

SSL (Secure Socket Layer) y TLS(Transport Layer Secure)

El protocolo SSL fue desarrollado por Netscape para permitir confidencialidad y autenticación en Internet. SSL es una capa por debajo de HTTP y tal como lo indica su nombre esta a nivel de socket por lo que permite ser usado no tan solo para proteger documentos de hipertexto sino también servicios como FTP, SMTP, TELNET entre otros.

La idea que persigue SSL es encriptar la comunicación entre servidor y cliente mediante el uso de llaves y algoritmos de encriptación.

El protocolo TLS esta basado en SSL y son similares en el modo de operar.

Es importante señalar que ambos protocolos se ejecutan sobre una capa de transporte definida, pero no determinada. Esto indica que pueden ser utilizados para cualquier tipo de comunicaciones. La capa de transporte más usada es TCP sobre la cual pueden implementar seguridad en HTTP.

Como punto de diferencia se puede mencionar que existen protocolos implementados sobre la capa de red, por ejemplo sobre IP. Tal es el caso de IPSec.

¿De que están compuestos?

Estos protocolos se componen de dos capas: el Record Protocol y el Handshake Protocol.

El Record Protocol es la capa inmediatamente superior a TCP y proporciona una comunicación segura. Principalmente esta capa toma los mensajes y los codifica con algoritmos de encriptación de llave simétrica como DES, RC4 aplicándole una MAC (Message Authentication Code) para verificar la integridad, logrando así encapsular la seguridad para niveles superiores.

El Handshake protocol es la capa superior a la anterior y es usada para gestionar la conexión inicial.

¿Cómo funcionan?

En resumidas cuentas, después que se solicita una comunicación segura, servidor y el cliente se deben poner de acuerdo en como se comunicaran (SSL Handshake) para luego comenzar la comunicación encriptada. Luego de terminada la transacción, SSL termina.

Solicitud de SSL:

Típicamente este proceso ocurre en el momento que un cliente accede a un servidor seguro, identificado con "https://...". pero como se mencionó, no necesariamente es usado para HTTP. La comunicación se establecerá por un puerto distinto al utilizado por el servicio normalmente. Luego de esta petición, se procede al SSL Handshake.

SSL Handshake:

En este momento, servidor y cliente se ponen de acuerdo en varios parámetros de la comunicación. Se puede dividir el proceso en distintos pasos:

- **Client Hello:** El cliente se presenta. Le pide al servidor que se presente (certifique quien es) y le comunica que algoritmos de encriptación soporta y le envía un número aleatorio para el caso que el servidor no pueda certificar su validez y que aun así se pueda realizar la comunicación segura.
- **Server Hello:** El servidor se presenta. Le responde al cliente con su identificador digital encriptado, su llave pública, el algoritmo que se usará, y otro número aleatorio. El algoritmo usado será el más poderoso que soporte tanto el servidor como el cliente.
- **Aceptación del cliente:** El cliente recibe el identificador digital del servidor, lo desencripta usando la llave pública también recibida y verifica que dicha identificación proviene de una empresa certificadora segura. Luego se procede a realizar verificaciones del certificado (identificador) por medio de fechas, URL del servidor, etc. Finalmente el cliente genera una llave aleatoria usando la llave pública del servidor y el algoritmo seleccionado y se la envía al servidor.
- **Verificación:** Ahora tanto el cliente y el servidor conocen la llave aleatoria (El cliente la generó y el servidor la recibió y desencriptó con su llave privada). Para asegurar que nada ha cambiado, ambas partes se envían las llaves. Si coinciden, el Handshake concluye y comienza la transacción.

Intercambio de Datos:

Desde este momento los mensajes son encriptados con la llave conocida por el servidor y el cliente y luego son enviados para que en el otro extremo sean desencriptados y leídos.

Terminación de SSL

Cuando el cliente abandona el servidor, se le informa que terminara la sesión segura para luego terminar con SSL.

En el siguiente esquema se muestra todo el proceso del Handshake:

insert esquema1

¿Cómo instalar un servidor seguro?

En primer lugar se debe disponer de un servidor que soporte SSL que en la actualidad es muy común en el mercado de los servidores. En caso de adquirir uno nuevo, se puede visitar [WebServer Compare](http://webcompare.internet.com) [http://webcompare.internet.com] para elegirlo.

Luego se debe elegir una empresa certificadora que respalde nuestra identificación. Una de las empresas más reconocidas es [VeriSign](http://www.verisign.com) [http://www.verisign.com] .

Luego de llenar formularios en la empresa elegida, esta entidad, después de unos días, envía el Certificado de Servidor Seguro por correo, el cual se debe guardar en un archivo de texto (incluyendo -----BEGIN CERTIFICATE----- y -----END CERTIFICATE-----)

Finalmente se siguen las instrucciones enviadas junto con el certificado y después de esto, al reiniciar el servidor, estará en condiciones de entablar comunicaciones seguras.