Secure Socket Layer, Transport Layer Security
SSL/TLS

Secure Socket Layer SSL es un protocolo diseñado para proveer:

- Privacidad
- Integridad

de los datos entre dos partes comunicantes.

Antecedentes:

SSL se envió como un borrador al cuerpo de ingenieros de Internet (Internet Engineering Task Force IETF) y se convirtió en el estándar RFC2246 (Transport Layer Security TLS 1.0)[1].

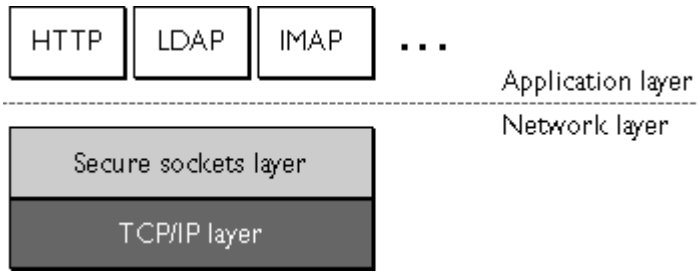

Figura 1: Ubicación de Secure Sockets Layer.

Objetivos Generales:

El protocolo TLS se ha diseñado para evitar que las aplicaciones cliente/servidor:

- Sean espiadas.
- Intromisiones.
- Falsificación.

Fases:

- Handshake.
- Transmisión de datos de la aplicación.

Handshake negocia:

- Algoritmos de cifrado.
- Claves.
- Autenticación del servidor y cliente.

Transmisión de datos:

- Una vez completado el handshake, se inicia la transmisión de los datos de la aplicación.

- Se cifran todos los datos utilizando las claves de la sesión negociadas durante el handshake.

Restricciones de exportación:

- Para versiones fuera de Estados Unidos de América se exporta una versión de SSL/TLS que utiliza claves de 40 bits, en E.U.A. se comercializa una versión de 128 bits.

Modelo de capas de TLS:

El protocolo está compuesto de 2 capas:

- TLS record protocol
- TLS handshake protocol

En su nivel más bajo, TLS record protocol se soporta sobre TCP.

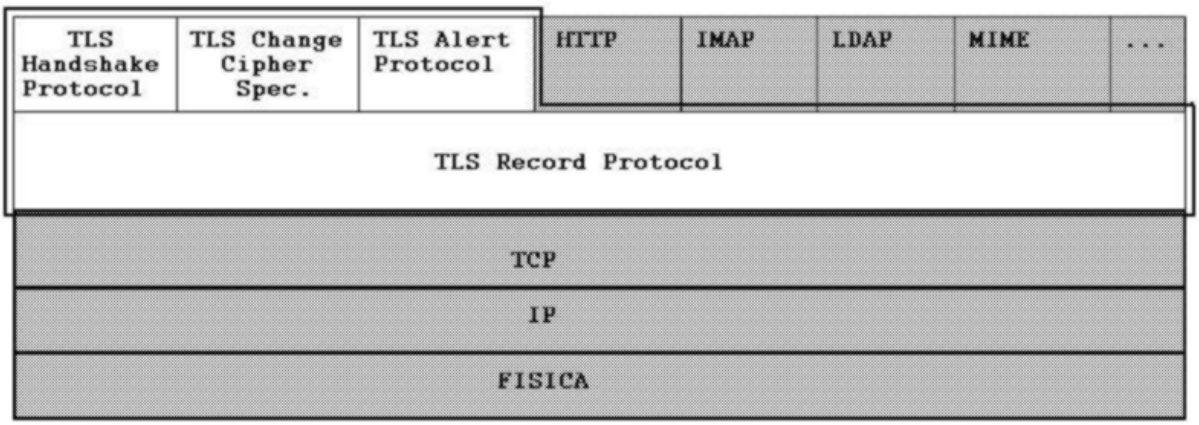

Figura 2. Capas de TLS.

Privacidad:

- Se usa criptografía simétrica para cifrado de datos (p.ej.: DES, RC4, etc.)
- Las claves para este cifrado simétrico se generan de forma única para cada conexión.
- Se basa en un secreto negociado por otro protocolo (p.ej.: TLS handshake protocol)
- Se puede utilizar sin cifrado.

Integridad:

- El transporte del mensaje incluye verificación de integridad del mensaje utilizando código de autenticación del mensaje con clave.
- El cálculo del código de autenticación del mensaje MAC se realiza mediante funciones de hash (p.ej.:, SHA, MD5, etc.)
- Puede operar sin un MAC, pero solo se usa en esta modalidad mientras otro protocolo esté utilizando el protocolo de registro como protocolo de transporte para negociar parámetros de seguridad.

Se puede utilizar TLS record protocol para encapsular varios protocolos de nivel más alto.

TLS handshake protocol:

- Permite autenticar el servidor y el cliente.

- Negocia el algoritmo cifrador y claves criptográficas antes que el protocolo de aplicación transmita o reciba su primer byte de datos.

Propiedades del protocolo de handshake:

- La identidad del punto se puede autenticar usando criptografía asimétrica o clave pública (p.ej.: RSA, DSS, etc.) Se requiere para al menos uno de los extremos.

- La negociación del secreto compartido es segura: el secreto negociado no esta disponible para espías, y tampoco se puede obtener el secreto con cualquier conexión auntenticada, aún por un atacante que pueda ubicarse en el medio de la conexión.

- La negociación es confiable: ningún atacante puede modificar la negociación de la comunicación sin ser detectado por las partes comunicantes.

Ventajas:

- TLS es independiente del protocolo de aplicación.

- Los protocolos de alto nivel pueden colocarse transparentemente sobre la capa TLS.

- Las desiciones de iniciar TLS handshaking y como interpretar los certificados de autenticidad se dejan a juicio de los diseñadores e instaladores de los protocolos que funcionan sobre TLS.

Objetivos:

- **Seguridad Criptográfica:** TLS debe utilizarse para establecer una conexión segura entre las dos partes.

- **Interoperabilidad:** los programadores independientes pueden desarrollar aplicaciones que utilicen TLS y podrán intercambiar parámetros criptográficos sin conocimiento del código del otro.

- **Extensibilidad:** TLS busca proveer de una malla en la que nuevos algoritmos de clave pública y de cifrado en bloque se puedan incorporar de ser necesario, con esto se evita la necesidad de crear un nuevo protocolo, y evita la necesidad de crear una nueva biblioteca de seguridad.

- **Eficiencia:** las operaciones criptográficas tienden a consumir recursos de CPU, particularmente las operaciones de clave pública, por esta razón el protocolo TLS incorpora un esquema de caché de sesión para reducir el número de conexiones que necesitarían establecerse desde cero. También se busca reducir la actividad de la red.

Diferencias con SSL versión 3

- TLS se basa en el desarrollo previo de SSL versión 3.0.
- Son muy similares pero no interoperan.
- TLS se identifica como una nueva versión de SSL 3.1.

HMAC y PRF

HMAC se denomina a la operación de construcción del código de autenticación del mensaje MAC.

En el handshake con dos algoritmos diferentes: MD5 y SHA-1.

PRF Pseudo Random Function toma como entrada un secreto, una semilla, y una etiqueta de identificación y produce una salida de longitud arbitraria.

La función PRF de TLS se crea dividiendo el secreto en dos mitades, una mitad para generar datos con P_MD5 y la otra mitad para generar datos con P_SHA-1, luego realiza la operación XOR con las salidas de estas dos operaciones de expansión juntas.

El protocolo de registro de TLS (TLS Record Protocol)

Es un protocolo de capas. En cada capa los mensajes incluyen los campos:

- Longitud.
- Descripción.
- Contenido.

Funciones de la capa de registro:

- Toma los mensajes a ser transmitidos.
- Fragmenta los datos en bloques manejables.
- Opcionalmente comprime los datos.
- Aplica un MAC.
- Cifra.
- Transmite el resultado.

Los datos recibidos son:

- Descifrados.
- Verificados.
- Descomprimidos.
- Reensamblados.
- y entregados a aplicaciones clientes de alto nivel.

Clientes del protocolo de registro (4):

- El protocolo de handshake.
- El protocolo de alerta.
- El protocolo de cambio de especificación de cifrador.
- y el protocolo de datos de la aplicación.

Propiedades de la capa de resgitro:

- Soporta tipos adicionales.

- Ignora los tipos no soportados.

- Requiere diseño cuidadoso contra ataques para cualquier protocolo a ser utilizado sobre TLS.

Estados de Conexión de la capa de registro

El estado de conexión TLS es el ambiente operativo del protocolo de registro de TLS.

Especifica un algoritmo de compresión, algoritmo cifrador, y algoritmo de código de autenticación del mensaje.

Los parámetros de estos algoritmos son:

- El secreto MAC.
- Las claves de cifrado de bloques
- Los valores de identificación de la conexión en las direcciones de lectura y escritura.

Existen siempre 4 estados lógicos de conexión importantes:

- Los estados activos de lectura y escritura.
- Los estados pendientes de lectura y escritura.

El protocolo de handshale puede activar selectivamente cualquiera de los estados pendientes, en este caso, el estado activo se reemplaza por el estado pendiente, y el estado pendiente se libera y reninicializa como vacio.

No es lícito crear un estado activo que no haya sido inicializado con los parámetros de seguridad correspondientes.

El estado inicial activo especifica que no utilizará cifrado, compresión o MAC.

Los parámetros de seguridad la cualquier estado son:

- Conexión terminal.
- La entidad puede cosiderarse la "cliente" o "servidora" de la conexión.
- Algoritmo cifrador de bloque.

En esta especificación se incluye el algoritmo utilizado para realizar el cifrado de bloque:

- El tamaño de la clave del algoritmo.
- Parte de la clave es secreta si se utiliza cifrado de bloque o de trama.
- El tamaño del bloque de cifrado (opcional) y si se trata de un cifrado exportable.

Capa de registro, Componentes:

- Algoritmo de código de autenticación de mensajes MAC.
- Algoritmo de compresión.
- Secreto maestro.(48 bytes compartido).
- Número aleatorio del cliente (32bytes).
- Número aleatorio del servidor (32 bytes)

La capa de registro utilizará los parámetros de seguridad para generar los siguientes 6 pasos:

- Secreto MAC de escritura del cliente.
- Secreto MAC de escritura del servidor.
- Clave de escritura del cliente.
- Clave de escritura del servidor.
- Vector de inicialización IV de escritura del cliente (para cifradores de bloque solamente).
- Vector de inicialización IV de escritura del servidor (para cifradores de bloque solamente).

El protocolo de handshake

Permite a los extremos:

- Acordar los parámetros de seguridad.
- Autenticarse.
- Instanciar los parámetros de seguridad negociados.
- Reportar condiciones de error.

Negociación:

- Identificador de sesión.
- Certificado de la parte interviniente. X509v3.
- Método de compresión
- Especificación del cifrador
- Secreto maestro
- Bandera reanudar

Protocolo de cambio de la especificación del cifrador

- Sirve para señalizar las transisiones entre estrategias de cifrado.

- El protocolo consiste de un solo mensaje, el que es cifrado y comprimido bajo el estado de conexión activo (no el pendiente) El mensaje consiste de un solo byte de valor 1.

- Se envia tanto al cliente como al servidor para notificar a la parte receptora que los registros subsiguientes serán protegidos bajo la nueva especificación y las nuevas claves de cifrado.

Protocolo de Alerta

Los mensajes de alerta expresan la severidad del mensaje y una descripción de la alerta.

Niveles de alerta:

- Fatal: inmediata finalización de la conexión
- A discreción: a discreción del implementador
- Advertencia: puede continuar pero se invalidad el identificador

Alerta de cierre

Cada parte debe intercambiar de mensajes notificación de cierre para evitar ataque por truncamiento.

Alertas de Error

Cuando se detecta un error, la parte que lo detecta envia un mensaje a la otra parte.

Hasta la transmisión o recibo de un mensaje de alerta fatal, ambas partes cierran inmediatemente la conexión.

Se requiere que Servidores y clientes olviden cualquier identificador de sesión, claves, y secretos asociados con una conexión fallida.

Alertas de Error

Alerta	Nivel
Mensaje inesperado	fatal
Mal registro MAC	fatal
Falla en descrifado	fatal
Desbordamiento registro	fatal
Falla descrompresión	fatal
Falla de handshake	fatal
Certificado malo	a discreción
Certif. no soportado	a discreción
Certificado revocado	a discreción
Certificado expirado	a discreción
Certif. desconocido	a discreción
Parámetro ilegal	fatal
CA Desconocida	fatal
Acceso denegado	fatal
Error decodificación	fatal
Error de decifrado	a discreción
Restricción exportación	fatal
Versión del protocolo	fatal
Seguridad Insuficiente	fatal
Error Interno	fatal
Cancelado por usuario	advertencia
No renegociación	advertencia

Protocolo de Hanshake
Dialogos entre las partes

Cliente		**Servidor**
Hola del cliente	------->	Hola del servidor
		Certificado*
		Intercambio Clave del Servidor*
		Petición Certificado*.
	<-------	Hola Servidor Efectuado.
Certificado*		
IntercambioClaveCliente		
VerificacionCertificado*		
[CambioEspecif.Cifrador]		
Finalizado	------->	[Cambio Especif. Cifrador]
	<-------	Finalizado
Datos Aplicación	<------>	Datos Aplicación

Figura 3 - Flujo del mensaje para un handshake completo.

*Indica mensajes opcionales o dependientes de la situación y que por tanto no siempre se envian.

Cliente		**Servidor**
HolaCliente	------->	HolaServidor
		[CambioEspecifCifrador]
	<-------	Finalización
[CambioEspecifCifrador]		
Finalización	------->	
Datos de Aplicación	<------>	Datos de Aplicación

Figura 4 - Flujo de mensajes para un handshake abreviado.

8. Cáculos Criptográficos

TLS requiere se especifiquen:

- Un conjunto de algoritmos.
- Un secreto maestro.
- Los valores aleatorios del cliente y del servidor.

Se determinan por el conjunto de cifradores seleccionado por el servidor y revelado en el mensaje Hola del servidor:

- La autenticación.
- Cifrado.
- y algoritmos MAC.

El algoritmo de compresión se negocia en los mensajes Hola.

Los valores aleatorios se intercambian en los mensajes Hola.

Todo lo que falta es calcular el secreto maestro.

Calculo del secreto maestro

Se utiliza para convertir el secreto pre-maestro en secreto maestro.

El secreto pre-maestro se debe eliminar de memoria una vez el secreto maestro se haya calculado.

De 48 bytes del longitud. La longitud de el secreto pre-maestro variará dependiendo en el método de intercambio de claves.

RSA

Un secreto pre-maestro de 48 byte lo genera el cliente, cifrado bajo la clave pública del servidor, y enviado al servidor.

El servidor utiliza su clave privada para decifrar el secreto pre-maestro. Ambas partes luego convierten el secreto pre-maestro en el secreto maestro, como se especifica arriba.

Las firmas digitales RSA se realizan utilizando PKCS #1 bloque tipo 1. El cifrado de clave pública RSA se realiza utilizando PKCS#1 bloque tipo 2.

Diffie-Hellman

Se realiza un cálculo Diffie-Hellman convencional. La clave negociada (Z) se utiliza como secreto pre-maestro, y es convertida en el secreto maestro.

9. Conjuntos Cifradores Obligatorios

En la ausencia de un perfil de aplicación estándar, una aplicación que cumpla con TLS debe implementar el conjunto cifrador TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA

10. Protocolo de aplicación de datos

Los mensajes de aplicación de datos son llevados por lacapa de registro y son fragmentados, comprimidos y cifrados basados en el estado de la conexión actual. Los mensajes se tratan como datos transparentes a la capa de registro.

REFERENCIAS

[1] The TLS Protocol Version 1.0, Request for Comments: 2246, Network Working Group, T. Dierks
Certicom, C. Allen, January 1999. ftp://ftp.isi.edu/in-notes/rfc2246.txt

[2] Security: The SSL White pages, 1998 Netscape Corp., http://docs.iplanet.com