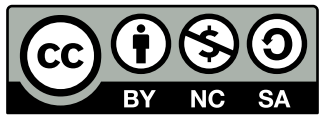
Seguridad en la Capa de Transporte (TLS)

Jesús Arias Fisteus

Aplicaciones Web (OpenCourseWare, 2023)

uc3m | Universidad Carlos III de Madrid
Departamento de Ingeniería Telemática



HTTP Secure (HTTPS)

Hypertext Transfer Protocol Secure (HTTPS) es una adaptación de HTTP para comunicaciones seguras, en la cual los mensajes se transmiten sobre el protocolo *Transport Layer Security* (TLS).

(El puerto TCP por defecto de HTTPS es 443)

Transport Layer Security (TLS)

Transport Layer Security (TLS) proporciona un canal de comunicación seguro entre dos pares.

- La conexión es **privada**, dado que los datos se cifran con *criptografía simétrica*.
- Se puede verificar la **identidad** del servidor mediante *la infraestructura de clave pública*.
- Se puede verificar la **integridad** de los datos, esto es, se pueden detectar pérdidas o alteraciones de los datos transmitidos.

Transport Layer Security (TLS)

- TLS es el estándar *de facto* para comunicaciones seguras entre aplicaciones.
- Tiene su origen en SSL (Secure Sockets Layer), desarrollado por primera vez en Netscape en los años 90.
- Es un estándar IETF.
- Su versión más actual es TLS 1.3 (RFC 8446).

Componentes del protocolo TLS

- **Protocolo "handshake":**
  - Establece una sesión TLS entre cliente y servidor.
- **Protocolo "record":**
  - Los datos se transmiten de forma segura del cliente al servidor y viceversa, en el contexto de la sesión TLS previamente establecida.

Pasos del protocolo "handshake"

1. **Negociación:**
   - Cliente y servidor acuerdan la versión de TLS, los algoritmos criptográficos y los parámetros a usar.
2. **Intercambio de claves:**
   - Cliente y servidor derivan las claves simétricas a usar para cifrar los datos transmitidos.
3. **Autenticación:**
   - El cliente autentica al servidor (y, opcionalmente, el servidor puede también autenticar al cliente).

“Handshake”: negociación

- El cliente especifica los algoritmos que puede usar enviando un mensaje **`ClientHello`**, el cual incluye:
  - Uno o más algoritmos de intercambio de claves.
  - Dos o más algoritmos de firma digital.
  - Una o más funciones de *hash* a usar para HKDF.
  - Uno o más algoritmos de cifrado con autenticación.
- El servidor escoge los algoritmos a usar de la lista de algoritmos especificados por el cliente, y se lo comunica a este enviando un mensaje **`ServerHello`**.

“Handshake”: intercambio de claves

1. Cliente y servidor comunican al otro sus claves públicas:
   - Las claves públicas que usen deben ser efímeras para proporcionar *forward secrecy*[1].
   - En TLS 1.3 estas claves pueden ser comunicadas en los mensajes `ClientHello` y `ServerHello` para agilizar el proceso.

[1] Secreto hacia adelante: la revelación de secretos utilizados en el intercambio de claves no compromete la seguridad de claves de sesión usadas en el pasado.

“Handshake”: intercambio de claves

2. Las claves simétricas que se usarán en la sesión se derivan mediante un algoritmo *HMAC-based Extract-and-Expand Key Derivation Function* (HKDF) y la función de *hash* seleccionada:
   - El cliente escoge un par de valores aleatorios y el servidor proporciona también aleatoriedad a las claves derivadas mediante su mensaje `ServerHello`.
   - En TLS 1.3 se derivan varias claves distintas para distintas partes de la comunicación.

(En TLS 1.3, una vez se han derivado las claves, todos los mensajes posteriores de la fase de “handshake” son cifrados.)

“Handshake”: autenticación

- Los servidores deben ser autenticados:
  - Los sitios Web necesitan un **certificado** (una clave pública de larga duración del sitio Web, firmada por una autoridad de certificación).
  - El certificado del sitio Web puede ser validado mediante la **infraestructura de clave pública (PKI)**.
- Opcionalmente, los clientes pueden ser también autenticados, aunque en la Web lo habitual es que solo se autentique a los servidores.

La infraestructura de clave pública

- Los navegadores Web confían en un conjunto de **autoridades raíz de certificación**.
- Las autoridades raíz de certificación pueden firmar certificados para **autoridades intermedias de certificación**. Estas, a su vez, pueden firmar los certificados de otras autoridades intermedias.
- Autoridades raíz e intermedias de certificación pueden firmar certificados para sitios Web.
- Se validan los certificados verificando su firma mediante el certificado de la autoridad de certificación que lo haya firmado. Se continúa sucesivamente con la verificación del certificado de cada autoridad de certificación intermedia en la **cadena de certificados** hasta alcanzar una entidad raíz.

Certificados

- Se usan normalmente certificados X.509.
- Los certificados de sitios Web contienen, entre otros:
  - La clave pública del sitio Web.
  - El o los nombres de dominio para los cuales aplica el certificado.
  - Fechas inicial y final de validez del certificado.
  - La firma de los datos del certificado por la autoridad de certificación que lo haya emitido.

“Handshake”: autenticación

1. El servidor envía:
   - Su propio certificado.
   - La cadena de certificados (lista de certificados de las autoridades de certificación que firman, en cadena, cada certificado, hasta alcanzar una autoridad raíz).
   - Una firma de todos los mensajes de “handshake” previos realizada con la clave secreta asociada a la clave pública del certificado del servidor.
2. El cliente verifica las firmas de los mensajes de “handshake” y de la cadena de certificados.

“Handshake”: reanudación de sesiones

- Es habitual que los clientes se conecten varias veces al mismo sitio Web, pero llevar a cabo un “handshake” en cada una de ellas es costoso.
- TLS permite reanudar sesiones previas reutilizando claves intercambiadas anteriormente mediante el **pre-session keys (PSK) handshake**:
  - Los servidores pueden producir un *ticket de sesión* que el cliente puede utilizar en el futuro para reanudar la sesión.
  - Los clientes pueden enviar el *ticket de sesión* en el mensaje `ClientHello`, y el servidor puede decidir en ese caso que se omitan la negociación, el intercambio de claves y la autenticación.

Transmisión de datos de aplicación (protocolo "record")

- **Cifrado de los datos de aplicación:**
  - Los datos son cifrados con una clave simétrica.
- **Protección frente a la manipulación de mensajes:**
  - Los mensajes son autenticados y la integridad de su contenido verificada mediante el uso de cifrado con autenticación.
- **Protección frente a mensajes repetidos o desordenados:**
  - Los datos se dividen en registros, a los cuales se asigna un número de secuencia. Ambos extremos verifican los números de secuencia de los registros que reciben.

Referencias

- The Transport Layer Security (TLS) Protocol Version 1.3. IETF RFC 8446. August 2018.
- David Wong, *Real-World Cryptography*, Manning Publications (2021):
  - Chapter 9 (Secure Transport).