APACHE SERVER

SSL / TLS

Salvador Mira

IES Pere Maria Orts

HTTPS

SSL es un protocolo criptográfico de comunicación utilizado para garantizar la identidad y la privacidad de las comunicaciones web

Técnicamente, **SSL** corresponde con la **capa de transporte** de **TCP/IP**

TLS (Transport Layer Security) es la siguiente generación de SSL: permite y garantiza el intercambio de datos en un entorno securizado y privado entre el cliente y el servidor, mediante aplicaciones como HTTP, POP3, IMAP, SSH, SMTP o NNTP

HTTPS

Hoy en día, **solo se debe utilizar TLS**

SSL 2.0 y **SSL 3.0** están desactualizados y ya no se consideran seguros

Lo mismo se aplica a las versiones más antiguas de **TLS**

- Solo se debe utilizar TLS 1.2 bajo ciertas condiciones indicadas en la especificación de TLS 1.3
- **Se debe evitar todos los protocolos SSL** y las versiones **1.0** y **1.1** de **TLS** (cuyo soporte se eliminará pronto)

Aunque a día de hoy se emplea **TLS**, es frecuente hacer referencia a **SSL**

SSL

FUNCIONAMIENTO

El proceso tiene **cinco fases**:

1. El **cliente envía** un mensaje de tipo **ClientHello**, que contiene información para el *HandShake*. Incluye:
   - Algoritmos de cifrado soportados
   - Versión máxima de **SSL** soportada
   - Métodos de compresión de datos
   - Un conjunto de bytes aleatorios que serán usados más adelante en un *Challenge* (reto)

2. El **cliente recibe** un mensaje de tipo **ServerHello** enviado por el **servidor**, que indica qué parámetros de los indicados por el cliente, se han seleccionado para la comunicación segura. Incluye el número de versión de **SSL**, la **configuración de cifrado** y **datos específicos de la sesión**. También recibe la clave pública del servidor

SSL

FUNCIONAMIENTO

El proceso tiene **cinco fases**:

3. El cliente autentica el certificado de servidor, por ejemplo, nombre común/ fecha/ emisor. El cliente (según el cifrado) crea el número secreto principal preliminar de la sesión, cifra con la clave pública del servidor y envía el número secreto principal preliminar cifrado al servidor
4. Descifrado y número secreto principal. El servidor utiliza su clave privada para descifrar el número secreto principal preliminar. El servidor y el cliente efectúan pasos para generar el número secreto principal con el cifrado acordado
5. El cliente y el servidor intercambian mensajes para informar de que se cifrarán futuros mensajes

SSL

Los **certificados SSL**:

- proporcionan una vía para **encriptar** el tráfico de **datos entre servidor y cliente**
- permiten **identificar el servidor web** y dar detalles a los visitantes acerca de la misma
- existen **autoridades de certificación** que los navegadores reconocen: sus certificados no proporcionan dudas
- los certificados **autofirmados** se pueden instalar en los servidores web, pero los navegadores no los reconocerán

SSL

Un certificado digital contiene un par de claves **privada** y **pública**

Está firmado por una **Autoridad Certificadora** (CA) que da veracidad de la identificación de la persona/empresa que está usando el certificado

La **Autoridad Certificadora** genera y firma el certificado para que los navegadores web sepan que ese certificado es de confianza

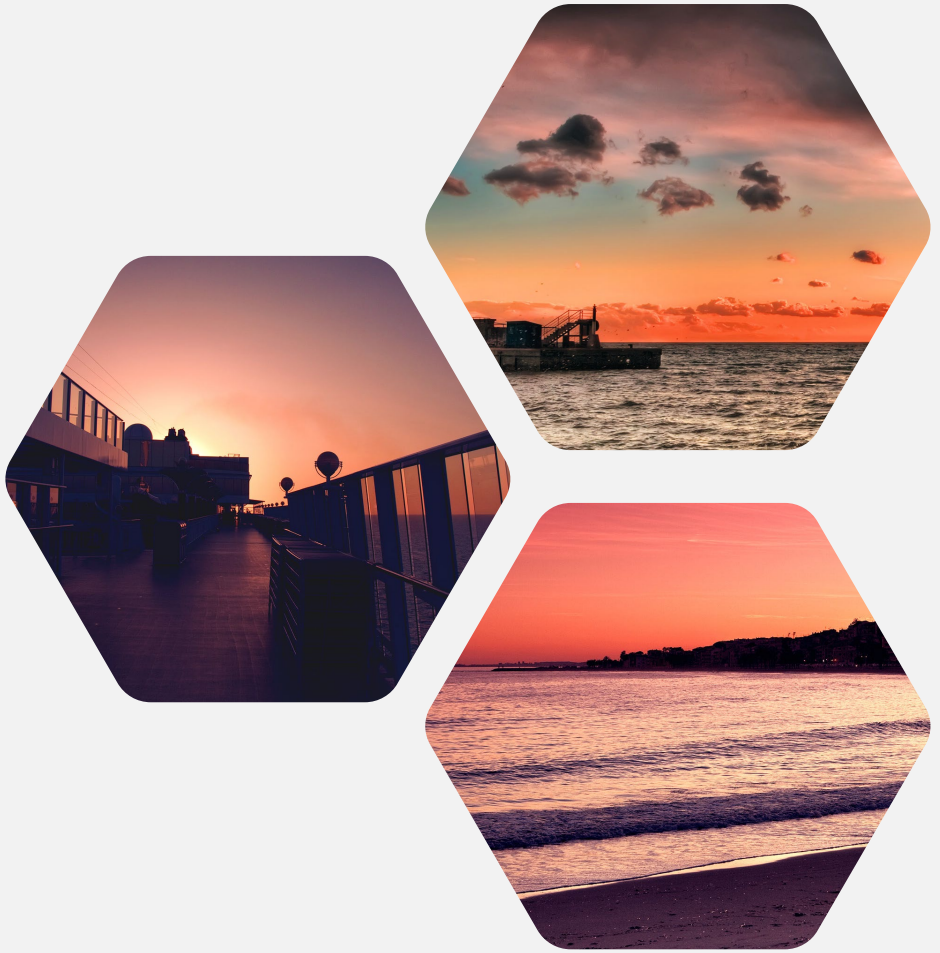
SSL

DOCUMENTO CSR

Cuando vamos a solicitar un **certificado SSL** a una **CA**, tenemos que crear y enviar a la misma un documento **CSR** (Certificate Signing Request)

En los **CSR** se incluye la siguiente información:

- **CommonName. Nombre del dominio**. Los certificados más básicos sólo permiten un único nombre de dominio, pero se pueden adquirir certificados multidominio y también *wildcard* para múltiples hosts de un dominio, *.dominio.tld
- **OrganitzationName. Nombre legal** exacto de la empresa u organización

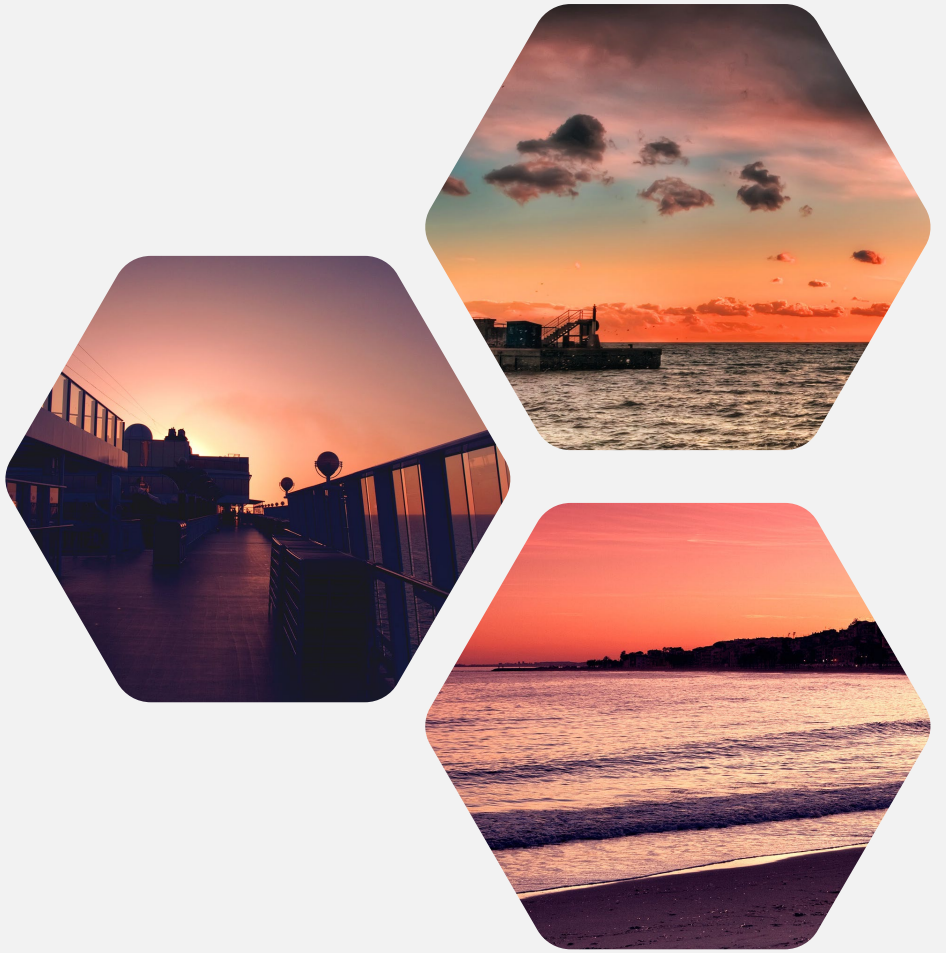


SSL

DOCUMENTO CSR

En los **CSR** se incluye la siguiente información:

- **OrganizationalUnit. Unidad organizativa de la empresa u organismo que pide el certificado**
- **CityOrLocality**. Ciudad en la que la empresa solicitante está ubicada legalmente
- **StateOrProvince**. Estado o provincia en el que la empresa se ubica legalmente
- **Country**. Abreviación ISO con dos letras que identifica al país en el cual se encuentra la empresa que pide el certificado. Ejemplo: **ES**



SSL

ESTÁNDARES EN LOS CERTIFICADOS

La **forma del certificado** así como la información que se incluye en él viene determinado por el **estándar X.509** del organismo de estandarización **ITU-T**

Existen distintos formatos para almacenar un certificado, los cuales se asocian a algunas extensiones típicas en los nombres de los archivos de los certificados

SSL

ESTÁNDARES EN LOS CERTIFICADOS

Formatos típicos de los certificados:

- **PEM (Privacy Enhanced Mail) y DER (Distinguish Encoding Rules):**
  - Son certificados que **incluyen la clave pública** pero **no la clave privada** (esta última se almacena en otro archivo de extensión «**.key**«)
  - Las extensiones de archivo más comunes son: **.pem, .crt, .cer** y **.der**
  - La diferencia entre **PEM** y **DER** es su codificación: **DER** se codifica en **binario** y **PEM** se codifica en **ASCII**

SSL

ESTÁNDARES EN LOS CERTIFICADOS

Formatos típicos de los certificados:

- **PKCS#7**
  - Es un **paquete de certificados** (se almacenan varios certificados en el archivo)
  - **No** incluye la clave privada en él
  - El estándar se define en el **RFC 2315**
  - Las **extensiones** de archivo más comunes son: **.p7b** y **p7c**

SSL

ESTÁNDARES EN LOS CERTIFICADOS

Formatos típicos de los certificados:

- **PKCS#12**
  - Es un **paquete de certificados** y **claves privadas** (encriptadas bajo una contraseña maestra definida por el usuario)
  - Es el sucesor del formato **PFX** de **Microsoft**
  - Archivo para el intercambio de información personal
  - Las extensiones de archivo más comunes son: **.pfx** y **.p12**

CERTIFICADOS AUTOFIRMADOS EN APACHE2

SSL

CONFIGURACIÓN EN APACHE2

Paso 1: Habilitar el módulo de **SSL** y reiniciar Apache para que tenga efecto:

```
sudo a2enmod ssl
sudo systemctl restart apache2
```

Paso 2: Crear un directorio en el que se almacenen las claves y certificados. Por ejemplo dentro de **/etc/apache2**

```
sudo mkdir /etc/apache2/ssl
```

SSL

CONFIGURACIÓN EN APACHE2

Paso 3: Instalar los paquetes **openssl** y **ca-certificates**

```
sudo apt install openssl ca-certificates
```

Paso 4: Generar la **clave privada** en el directorio almacén de los certificados y establecer los **permisos** para que **sólo el usuario root** pueda acceder a ella

```
cd /etc/apache2/ssl
sudo openssl genrsa -out server.key 2048
sudo chmod 600 server.key
```

SSL

CONFIGURACIÓN EN APACHE2

Paso 5: Generar el **archivo CSR** (Certificate Signing Request)

```
sudo openssl req -new -key server.key \
-out server.csr
```

El comando nos pregunta de forma interactiva una serie de parámetros.

- El más importante es el nombre del sitio web "**Common Name**"
- Si no lo tenemos, indicamos la dirección IP

SSL

CONFIGURACIÓN EN APACHE2

Paso 5: Generar el **archivo CSR** (Certificate Signing Request)

```
administrador@debian-despliegue:/etc/apache2/ssl$ sudo openssl req -new -key ser
ver.key -out server.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Alacant
Locality Name (eg, city) []:Benidorm
Organization Name (eg, company) [Internet Widgits Pty Ltd]:DAW
Organizational Unit Name (eg, section) []:Informática
Common Name (e.g. server FQDN or YOUR name) []:olimpo.daw
Email Address []:smira@olimpo.daw

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

SSL

CONFIGURACIÓN EN APACHE2

Paso 6: Generar el **certificado autofirmado**

```
sudo openssl x509 -req -days 365 \
-in server.csr -signkey server.key \
-out server.crt
```

Paso 7: Crear un **virtual host** que haga uso del certificado y por tanto de SSL

- Podemos usar como plantilla el fichero /etc/apache2/sites-available/default-ssl.conf creando una copia

```
cd /etc/apache2/sites-available
sudo cp default-ssl.conf olimpo_ssl.conf
```

SSL

CONFIGURACIÓN EN APACHE2

Paso 7: Continuación…

- Modificamos el nombre del virtual host con el nombre de dominio usado para la generación del certificado
  <VirtualHost olimpo.daw:443>

- Nos aseguramos que las siguientes directivas están y que apuntan a los ficheros de claves apropiados

```
SSLEngine on
SSLCertificateFile /etc/apache2/ssl/server.crt
SSLCertificateKeyFile /etc/apache2/ssl/server.key
```

SSL

CONFIGURACIÓN EN APACHE2

Paso 8: Habilitar el virtual host mediante **a2ensite** y reiniciar el servidor

```
sudo a2ensite olimpo_ssl.conf
sudo systemctl restart apache2
```

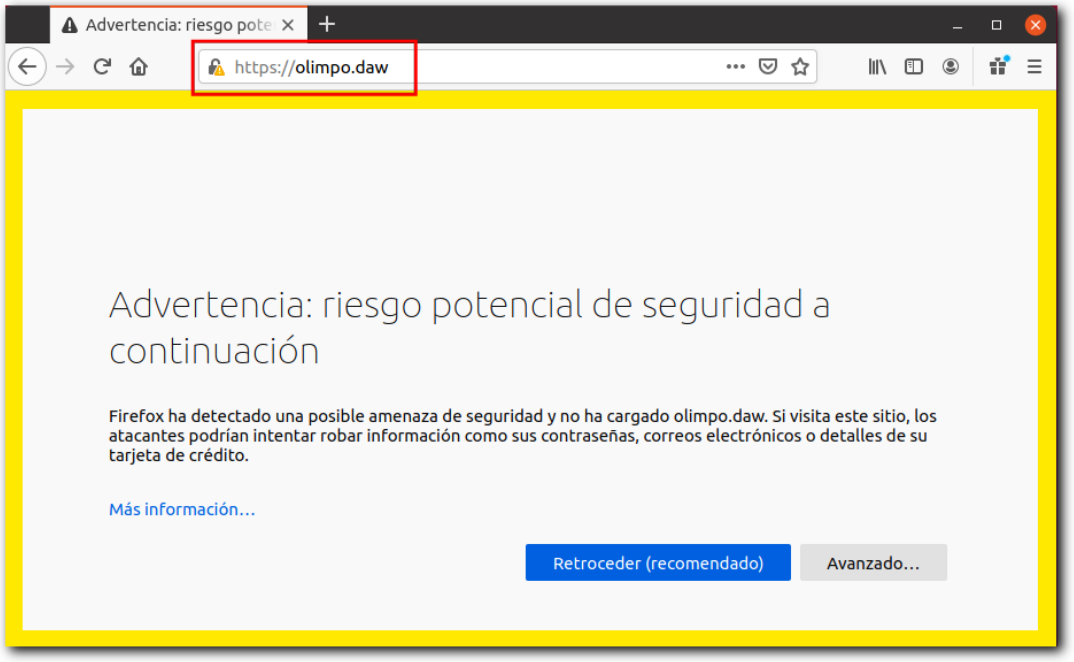
Paso 9: Visualizar el nuevo sitio con **https://nombre_sitio**

SSL

CONFIGURACIÓN EN APACHE2

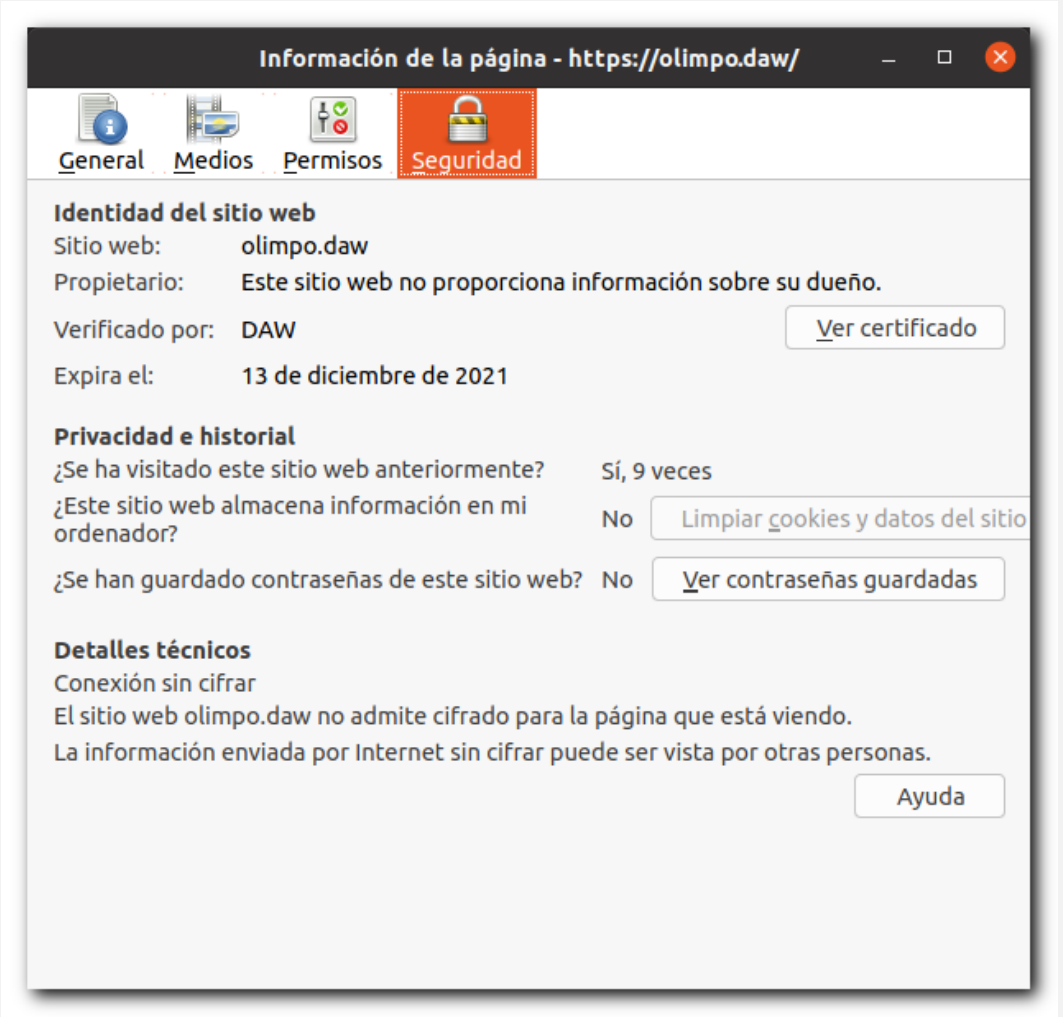
Al abrir la página securizada, el navegador nos da una **advertencia de seguridad**

Esto es debido a que el certificado es autofirmado y el navegador desconoce la **CA**



SSL

CONFIGURACIÓN EN APACHE2



SSL

CONFIGURACIÓN EN APACHE2

Hay que tener en cuenta que el certificado SSL sólo es válido para el nombre de servidor para el cual se generó

Si se tiene intención de usar para **múltiples nombres**, habrá que generarlo para soportar esta funcionalidad

Algo típico es utilizar **Wilcards** en el **CommonName** para soportar múltiples nombres en un mismo subdominio