CFGS DAW Semipresencial

MANUAL SSL
DAW

INSTALACCION CERTIFICADO SSL EN UBUNTU

FAJARDO MUÑOZ, JOSE
josfajmun@alu.edu.gva.es

INDICE

1. ¿Qué es SSL?

SSL o en sus siglas Secure Sockets Layer, o en su version más reciente TLS (Transport Layer Security) es un protocolo de seguridad en internet, que se encarga de proteger y cifrar la comunicación entre un servidor web y un navegador de un cliente de manera que no pueda ser intervenida y/o manipulada por terceros.

1.1 ¿Para qué sirve?

- Cifrado de datos: Codifica la información que se transmite entre el servidor y el navegador, de modo que es ilegible en caso de ser interceptada. (ataque MiTM)

- Autenticación del sitio web: Verifica la identidad del sitio web al que te conectas de modo que estas en el sitio web que dice ser y no una copia falsa (Phishing)

- Protección de datos sensibles: Sobre todo para datos confidenciales, personales, bancarios, contraseñas, etc…

1.2 ¿Cómo funciona?

- Solicitud de conexión: Cuando accedes a un sitio web seguro (HTTP**S**), tu navegador envía una solicitud al servidor.

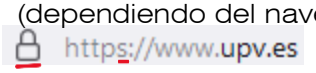
- Envío del certificado SSL: El servidor responde enviando su certificado SSL, que contiene su clave pública.

- Verificación del certificado: Tu navegador verifica la autenticidad del certificado SSL a través de una autoridad de certificación (CA).

- Creación de una conexión cifrada: Si el certificado es válido, se establece una conexión cifrada entre tu navegador y el servidor. Todas las comunicaciones posteriores se encriptan y desencriptan utilizando claves públicas y privadas.

1.3 ¿Por qué es importante SSL?

- Confianza del usuario: Los sitios web con certificado SSL aparecen en el navegador con un indicativo, o bien un candado, o de color verde (dependiendo del navegador) en la barra de direcciones y el protocolo HTTPS.



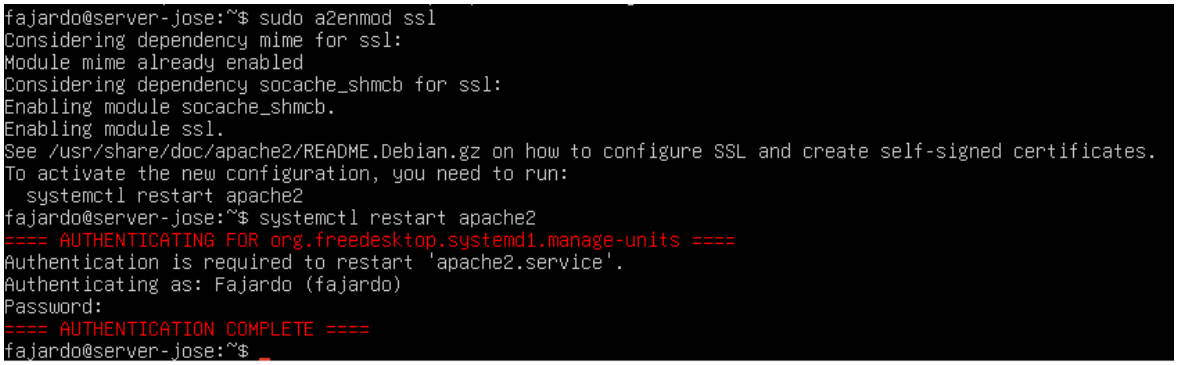
- Protección contra ataques: El cifrado de SSL dificulta que se intercepten y roben datos sensibles.

- Cumplimiento de normativas: Muchas industrias tienen requisitos legales para proteger los datos de los clientes, y SSL es una forma de cumplir con estas normativas.

2. Manual de instalación SSL

2.1 Requisitos previos

- Iniciar sesión en Ubuntu server como usuario habilitado con permisos de administrador.
- Apache instalado
- Y tener todos los paquetes actualizados con el comando: *"sudo apt update"* y *"sudo apt upgrade"*
- Habilitar mod_ssl para poder utilizar certificados SSL, es un modulo de apache que proporciona soporte para el cifrado SSL. *"sudo a2enmod ssl"* y reiniciar apache para activar el módulo con el comando *"sudo systemctl restart apache2".*



2.2 Generar una CSR (solicitud de certificado)

- Empleamos el comando *"sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/sslapache.key -out /etc/ssl/certs/sslapache.crt"*

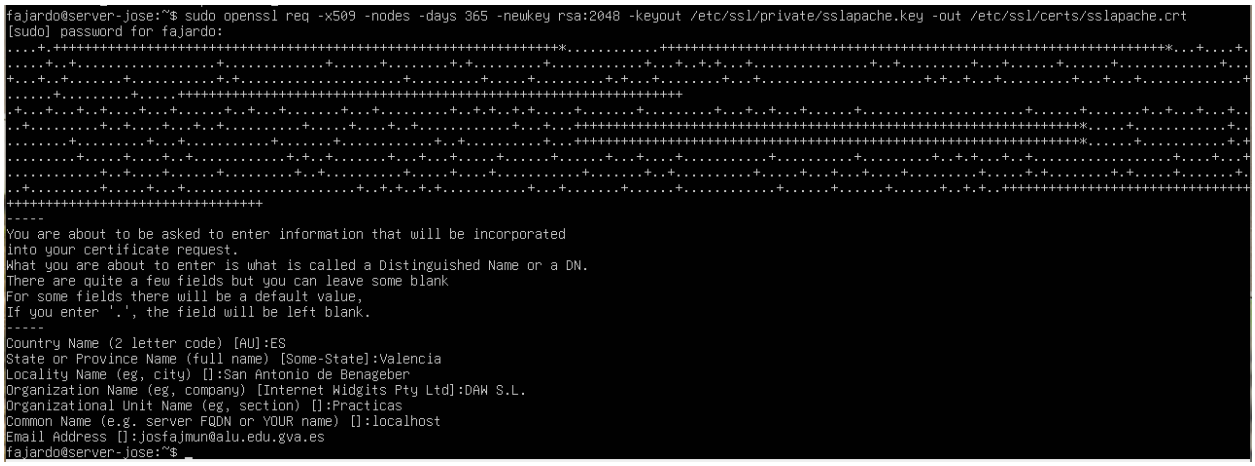
- A continuación, detallamos que significa cada una de las opciones:

  - ✓ openssl: es la herramienta de línea de comandos para crear y administrar certificados, claves y otros archivos de OpenSSL.
  - ✓ req -x509: especifica que deseamos usar la administración de la solicitud de firma de certificados (CSR) X.509. El "X.509" es un estándar de infraestructura de claves públicas al que se adhieren SSL y TLS para la administración de claves y certificados.
  - ✓ -nodes: indica a OpenSSL que omita la opción para proteger nuestro certificado con una frase de contraseña. Es necesario que apache pueda leer el archivo al iniciarse el servidor sin que el usuario tenga que introducir la contraseña, si no tendríamos que introducirla tras cada reinicio.
  - ✓ -days 365: esta opción establece el tiempo durante el cual el certificado se considerará válido. En este caso, lo configuramos por un año. Muchos navegadores modernos rechazarán cualquier certificado válido por más de un año.
  - ✓ -newkey rsa:2048: especifica que deseamos generar un nuevo certificado y una nueva clave al mismo tiempo. No creamos la clave que se requiere para firmar el certificado en un paso anterior, por lo que debemos crearla junto con el certificado. La parte rsa:2048 le indica que cree una clave RSA de 2048 bits de extensión.

- ✓ -keyout: esta línea indica a OpenSSL dónde colocar el archivo de clave privada generado que estamos creando.
- ✓ -out: indica a OpenSSL dónde colocar el certificado que creamos.

2.3 Obtener el Certificado SSL

- A continuación, nos solicitara unos datos para incorporar al certificado SSL. Rellenamos los datos requeridos.



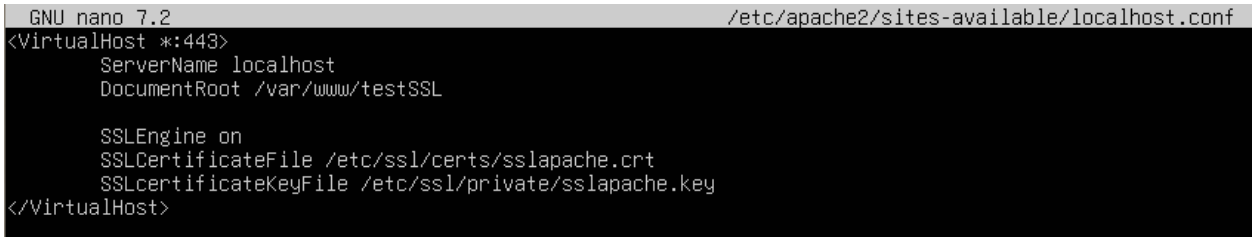
Ya tenemos los certificados y claves en las carpetas con los nombres indicados

2.4 Instalación del Certificado en Apache

Una vez tenemos nuestro certificado, nos dirigimos a la carpeta de configuración SSL del servidor local de apache.

"sudo nano /etc/apache2/sites-available/localhost.conf"

Añadimos la configuración mínima:



Ahora creamos la carpeta donde estará ubicada nuestra página web y el archivo que permitirá comprobar que funciona.

"sudo mkdir /var/www/testSSL"

"sudo nano /var/www/testSSL/index.html"

Introducimos en él, algo básico que muestre para comprobar que funciona como, por ejemplo:

"<h1>FUNCIONA!</h1>"

Guardamos, cerramos y habilitamos el archivo de configuracion:

"sudo a2ensite localhost.conf"

Comprobaremos que no hay errores de configuración:

"sudo apache2ctl configtest"

Si al final indica syntax Ok, no hay errores de sintaxis en la configuración.

Hacemos un reload de apache2 con:

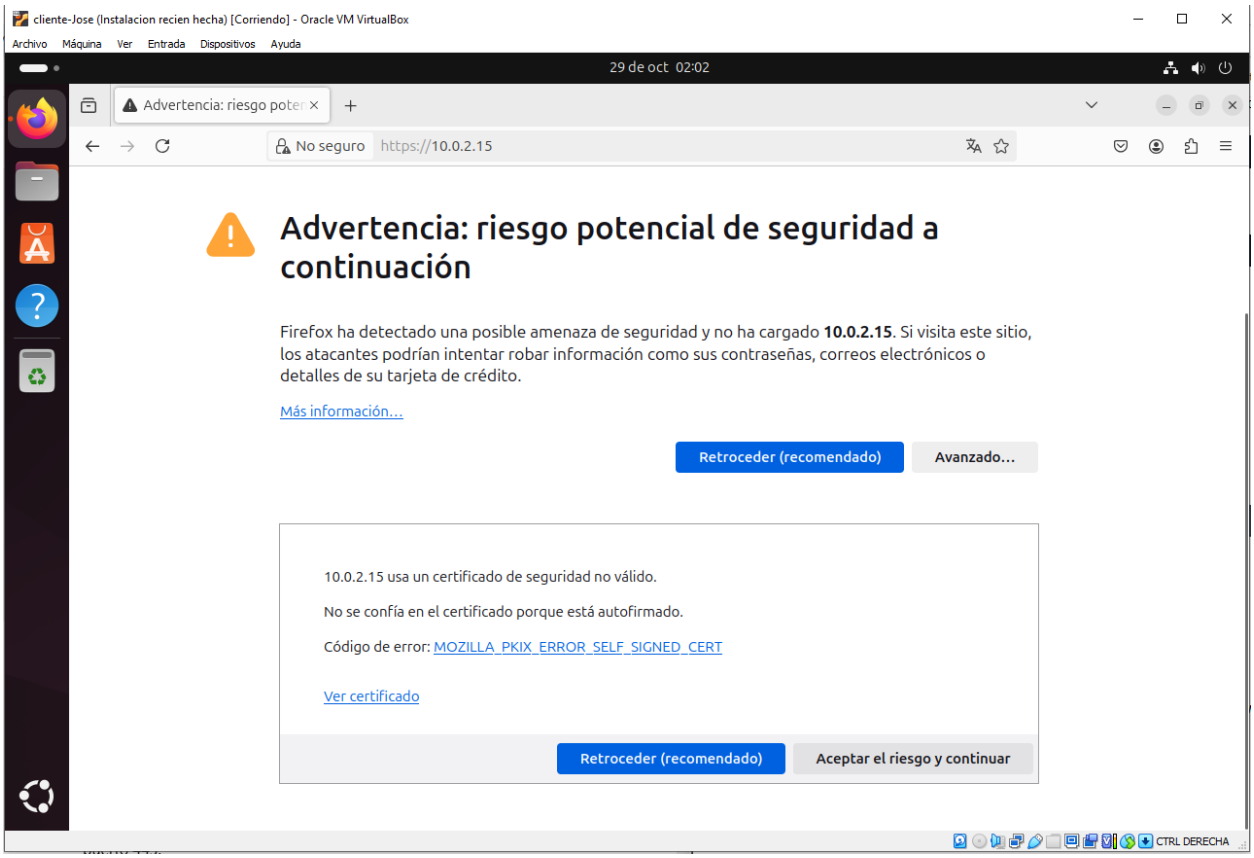
"sudo systemctl reload apache2"

```
fajardo@server-jose:/var/www/testSSL$ sudo a2ensite localhost.conf
Enabling site localhost.
To activate the new configuration, you need to run:
  systemctl reload apache2
fajardo@server-jose:/var/www/testSSL$ sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress th
is message
Syntax OK
fajardo@server-jose:/var/www/testSSL$ systemctl reload apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to reload 'apache2.service'.
Authenticating as: Fajardo (fajardo)
Password:
==== AUTHENTICATION COMPLETE ====
```

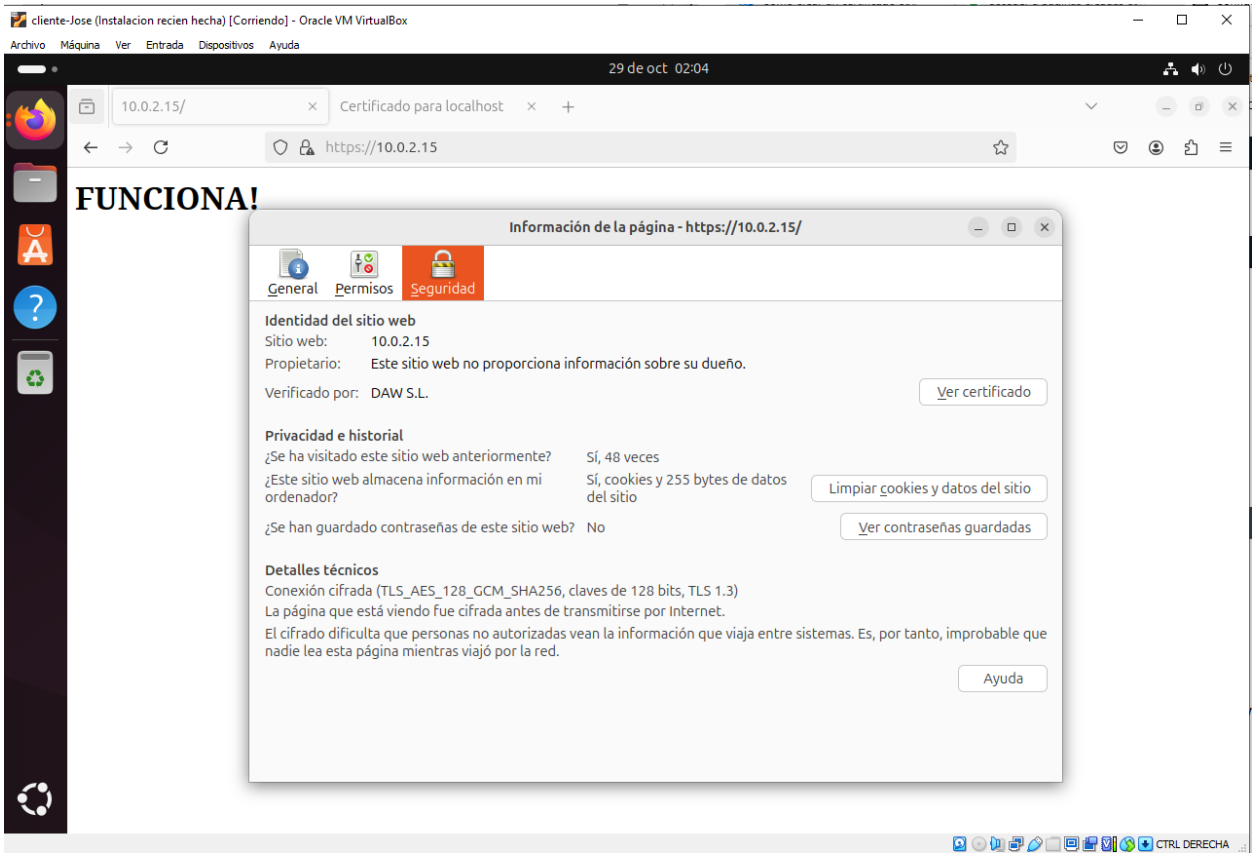
Arrancamos la máquina virtual con Ubuntu desktop para comprobar que funciona.

Accedemos al navegador e introducimos la IP del servidor. En mi caso 10.0.2.15, pero con el HTTPS delante.

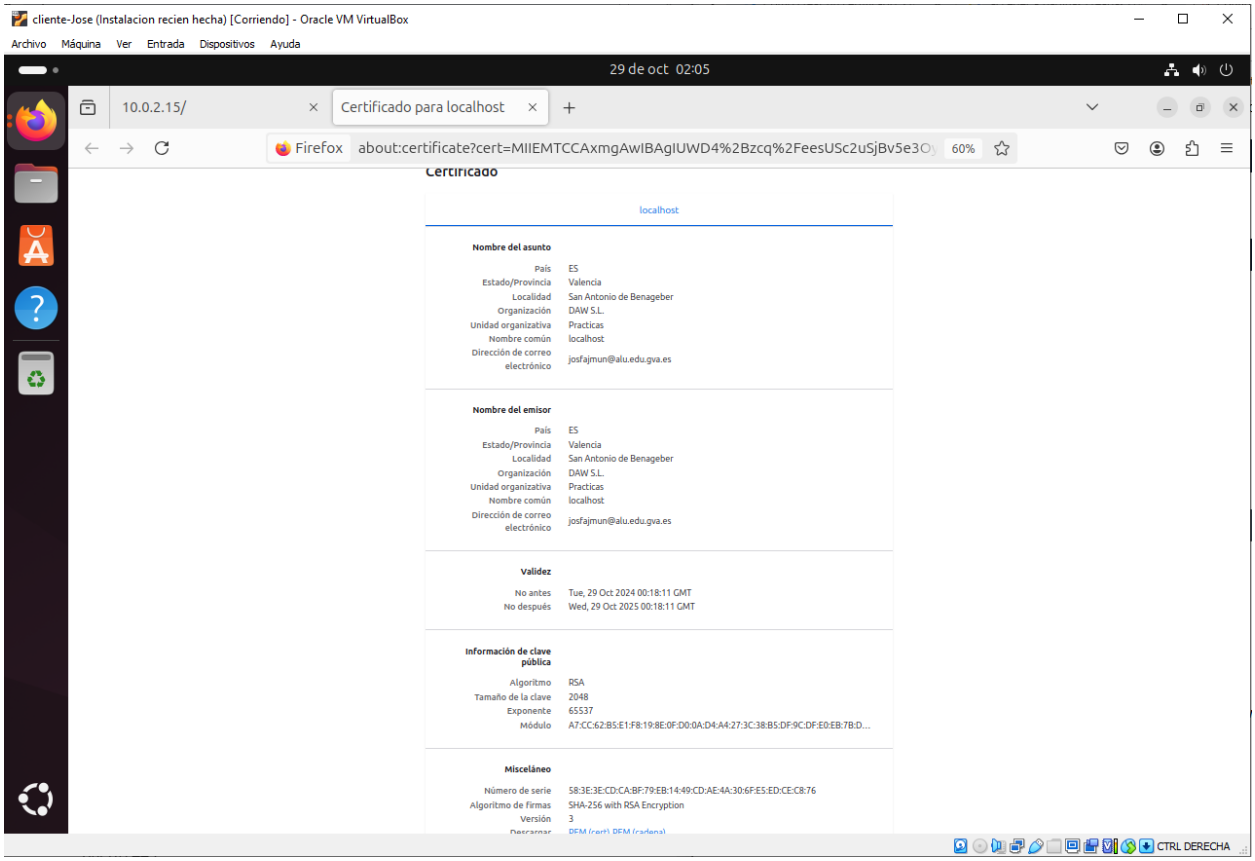
https://10.0.2.15



La advertencia que nos aparece no es ni mas ni menos, de que el certificado no esta validado por una entidad de certificación, si no que es auto firmado. Aceptamos el riesgo en este caso por que hemos generado el certificado nosotros mismos.



Si pinchamos en ver certificado, vemos los datos introducidos para la generación del certificado.



2.5 Redirigir las peticiones de http a https

Vamos a redirigir todas las peticiones de conexión al puerto 443 https. Para ello en nuestro localhost.conf añadimos lo siguiente:

"Redirect / https://localhost/"

```
<VirtualHost *:443>
        ServerName localhost
        DocumentRoot /var/www/testSSL

        SSLEngine on
        SSLCertificateFile /etc/ssl/certs/sslapache.crt
        SSLcertificateKeyFile /etc/ssl/private/sslapache.key
</VirtualHost>
<VirtualHost *:80>
        ServerName localhost
        Redirect / https://localhost/
</virtualHost>
```

Comprobamos que este correcta la sintaxis y recargamos apache.

"sudo apache2ctl configtest"

"sudo systemctl reload apache2"

```
fajardo@server-jose:/var/www/testSSL$ sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress th
is message
Syntax OK
fajardo@server-jose:/var/www/testSSL$ systemctl reload apache2
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to reload 'apache2.service'.
Authenticating as: Fajardo (fajardo)
Password:
==== AUTHENTICATION COMPLETE ====
fajardo@server-jose:/var/www/testSSL$
```

Ahora cada vez que accedemos a *http://10.2.0.15/index.html*, nos redirige a: *https://10.2.0.15/index.html*

3. Fuentes

Cloudflare. *¿Qué es SSL?*. https://www.cloudflare.com/es-es/learning/ssl/what-is-ssl/

DigiCert. *¿Qué es SSL, TLS y HTTPS?*. https://www.digicert.com/what-is-ssl-tls-and-https

Apache Software Foundation. *mod_ssl.* https://httpd.apache.org/docs/current/mod/mod_ssl.html

SSL Market. *Instalación del certificado SSL en el servidor Apache.* https://www.sslmarket.mx/ssl/instalacion-del-certificado-ssl-en-el-servidor-apache

DigitalOcean. *Cómo crear un certificado SSL autofirmado para Apache en Ubuntu 20.04.* https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-apache-in-ubuntu-20-04-es

Gigacore. *Cómo instalar un certificado SSL/TLS en Apache Open SSL.* https://aprende.gigacore.io/docs/instalacion/como-instalar-un-certificado-ssl-tls-en-apache-open-ssl/

SSL.com. *Códigos de país.* https://www.ssl.com/es/c%C3%B3digos-de-pa%C3%ADs