



# **Centro de Enseñanza Técnica Industrial**

## **Desarrollo de Software**

### **Instalación del servidor HTTP + SSL**

**Jesús Alberto Aréchiga Carrillo**

**22310439      5N**

**Profesor**

**José Francisco Pérez Reyes**

**Diciembre 2024**

**Guadalajara, Jalisco**

# Introducción

La implementación de un servidor HTTP con certificación SSL (Secure Sockets Layer) responde a la necesidad de garantizar la confidencialidad e integridad de la información intercambiada entre clientes y servidores a través de internet. Al integrar un certificado SSL, se establece un canal de comunicación cifrado, impidiendo que terceros no autorizados puedan leer o alterar datos sensibles. Este enfoque es fundamental en el desarrollo de aplicaciones web seguras, ya que refuerza la confianza del usuario y cumple con estándares internacionales de seguridad, evitando vulnerabilidades como la interceptación de datos, el robo de información o la manipulación de contenidos. De esta forma, el servidor HTTP con SSL se consolida como un componente esencial en entornos de producción, asegurando la protección de la comunicación y la integridad del ecosistema digital.

## Desarrollo

El objetivo principal de este proyecto es habilitar el certificado SSL en un servidor HTTP para hacer las conexiones seguras. En este caso se utiliza un servidor con una IP 10.0.0.9.

Para comenzar a instalar el certificado se utiliza el comando:

```
# openssl req -x509 -nodes -days 1825 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
```

La consola va a mostrar los campos del certificado que hay que ingresar para poder generarlo:

Country Name (2 letter code) [GB]: MX

State or Province Name (full name) [Berkshire]: Jalisco

Locality Name (eg, city) [Newbury]: Guadalajara

Organization Name (eg, company) [My Company Ltd]: Empresa, S.A. de C.V.

Organizational Unit Name (eg, section) []: Departamento de TI

Common Name (eg your name or your server's hostname) []: \*.dominio.org

Email Address []: webmaster@dominio.org

Se le dan los permisos al certificado y a la clave privada con

```
# chmod 400 /etc/apache2/ssl/apache.crt /etc/apache2/ssl/apache.key
```

Ahora se agrega el módulo a Apache:

```
# a2enmod ssl
```

Ahora se agregan los VirtualHost cambiando el puerto 80 por el puerto 443:

```
<VirtualHost *:443>
```

```
    ServerName www.practicahttps.com.mx
```

```
    ServerAdmin webmaster@localhost
```

```
    DocumentRoot /home/user
```

```
    SSLEngine On
```

```
    SSLCertificateFile /etc/ssl/certs/apache.crt
```

```
    SSLCertificateKeyFile /etc/ssl/private/apache.key
```

```
<Directory /home/user/>
```

```
    Require all granted
```


```
</Directory>
```

```
</VirtualHost>
```

Se reinicia el servicio y se prueba.

## Evidencias

Consola mostrando la dirección IP de la máquina virtual con la que está trabajando en ese momento.



```
a22310439@a22310439: ~
Password:
root@a22310439:~# l
root@http5:~# ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enX0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:16:3e:a5:94:b8 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.9/24 brd 10.0.0.255 scope global enX0
        valid_lft forever preferred_lft forever
    inet6 fe80::216:3eff:fea5:94b8/64 scope link
        valid_lft forever preferred_lft forever
root@http5:~#
```

Consola mostrando los procesos que se están ejecutando en la máquina virtual y visualizando correctamente el proceso apache2.

```
a22310439@a22310439: ~  
226 ?      00:00:00 cron  
227 ?      00:00:00 dbus-daemon  
229 ?      00:00:00 systemd-logind  
231 ?      00:00:00 cryptd  
346 tty1    00:00:00 agetty  
348 hvc0    00:00:00 login  
353 ?      00:00:00 sshd  
444 ?      00:00:01 apache2  
476 ?      00:00:00 apache2  
477 ?      00:00:00 apache2  
478 ?      00:00:00 apache2  
479 ?      00:00:00 apache2  
480 ?      00:00:00 apache2  
481 ?      00:00:00 apache2  
482 ?      00:00:00 apache2  
672 ?      00:00:00 kworker/u2:1-events_unbound  
679 ?      00:00:00 systemd  
681 ?      00:00:00 kworker/0:0-events  
683 ?      00:00:00 (sd-pam)  
698 hvc0    00:00:00 bash  
835 ?      00:00:00 kworker/0:2-cgroup_destroy  
836 ?      00:00:00 kworker/u2:0-events_unbound  
843 hvc0    00:00:00 ps  
root@http5:~#
```

Consola del Dom0 mostrando la información de la ejecución de la máquina virtual que está ejecutando el servidor de HTTPS.

```
a22310439@a22310439: ~  
xentop - 23:31:50 Xen 4.17.4-pre  
2 domains: 1 running, 1 blocked, 0 paused, 0 crashed, 0 dying, 0 shutdown  
Mem: 16658304k total, 16652336k used, 5968k free CPUs: 8 @ 2807MHz  
k) VBDs VBD_00 VBD_RD VBD_WR VBD_RSECT VBD_WSECT SSID  
Domain-0 -----r 361 6.2 15433356 92.6 no limit n/a 8 0 0  
0 0 0 0 0 0 0  
https --b--- 15 0.1 1048576 6.3 1049600 6.3 1 0 0  
0 2 0 4062 917 336034 20328 0  
Delay Networks vds CPUs Repeat header Sort order Quit
```

The screenshot shows a terminal window with a dark background. At the top, the browser's address bar displays "http://192.168.1.85". The main content area shows the output of the PHP function `phpinfo()`, which provides detailed information about the current PHP installation. The title of the page is "PHP Version 8.2.26". The output is organized into sections, with "System" being the first and most prominent one. It lists various system details such as the operating system (Linux), build date, server architecture (Apache), and configuration files. A large portion of the "System" section is dedicated to listing loaded modules and their paths, including GD, MySQL, Redis, and various database drivers. Other sections visible include "PHP API", "PHP Extensions" (listing Zend-related extensions like Zend OPcache and Zend Session), "Zend Engine Build", "PHP Extension Build", "Debug Build", "Thread Safety", "Signal Handling", "Memory Manager", "Multibyte Support", "Execution Timers", "IPv6 Support", "Dynamic Shared Object (DSO) Support", "Registered PHP Streams", "Registered Stream Socket Transports", and "Registered Stream Filters".

La implementación de un servidor HTTP con un certificado SSL resulta esencial para salvaguardar la integridad y la confidencialidad de la información transmitida en entornos digitales. Al cifrar las comunicaciones, se previenen ataques como la interceptación, manipulación y robo de datos, garantizando así una experiencia más segura para los usuarios. Además, el uso de un certificado SSL impulsa la confianza en el servicio, contribuyendo a la reputación de la organización y al cumplimiento de estándares y normativas internacionales. En definitiva, el despliegue de un servidor HTTP protegido con SSL se ha consolidado como una práctica fundamental en la arquitectura de sistemas web modernos, sentando las bases para una interacción segura, confiable y transparente en el ecosistema digital.