

SSL EN APACHE Y MÓDULOS



DAVID SEGURA TRISTANCHO
2º ASIR 2022/23

ÍNDICE

1.	Introducción	2
2.	Configuración SSL.....	2
3.	Certificado Digital SSL.....	4
4.	Módulo Rewrite y Redirect	5
5.	Módulo Userdir.....	7

1. INTRODUCCIÓN

En las siguientes páginas vamos a indagar en algunos de los módulos, de entre los muchos que existen, que atesora Apache y que resultan de enorme utilidad. Vamos a comprobar sus diversas funciones en un Ubuntu Server, y vamos a comenzar en primer término por SSL.

SSL es una tecnología estandarizada que permite cifrar el tráfico de datos entre un navegador web y un sitio web (o entre dos servidores web), protegiendo así la conexión. Para ello, vamos a ver su funcionamiento tomando como ejemplo un sitio creado para ello (sitio4).

2. CONFIGURACIÓN SSL

Comenzamos creando, por lo tanto, el sitio4 dentro de /var/www. A su vez, dentro de sitio4 creamos el archivo index.html.

```
usuario@usuario:/var/www/sitio4$ ls  
index.html
```

```
GNU nano 4.8 index.html  
Bienvenidos a la practica 4 del tema de Apache, SSL y diferentes modulos.
```

Dentro de los sitios disponibles, creo un archivo de configuración para el sitio4 y lo activo. Desactivo el archivo por defecto y habilito el de nuestro sitio. Ahí no queda el asunto, para permitir el uso de ssl habré de habilitar el modulo encargado de ello.

```
usuario@usuario:/etc/apache2/mods-available$ sudo a2enmod ssl
```

Cuando lo habilite, crearé en sitios disponibles un archivo de configuración ssl para el sitio4.

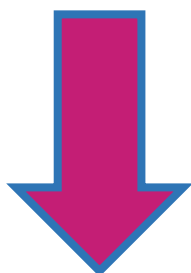
```
usuario@usuario:/etc/apache2/sites-available$ sudo cp default-ssl.conf sitio4ssl.conf
```

Lo activo y si me dirijo dentro de apache2 a los sitios activados tendremos tanto el de sitio4 para el puerto 80 (sitio4.conf), como el creado para el puerto 443 (sitio4ssl.conf).

```
usuario@usuario:/etc/apache2/sites-available$ sudo a2ensite sitio4ssl.conf
```

```
usuario@usuario:/etc/apache2/sites-enabled$ ls  
sitio4.conf sitio4ssl.conf
```

Si echamos un vistazo al archivo de configuración del sitio4 nos cercioraremos de que está habilitado para cualquier interfaz y el puerto 80. Hecho cada cambio reiniciamos el servicio para aplicar tales modificaciones con `sudo service apache2 restart`.



```
GNU nano 4.8                               sitio4.conf
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn
```

Y si echamos un vistazo al archivo de configuración de nuestro sitio versión ssl lo tendremos por defecto predeterminado para el puerto 443.

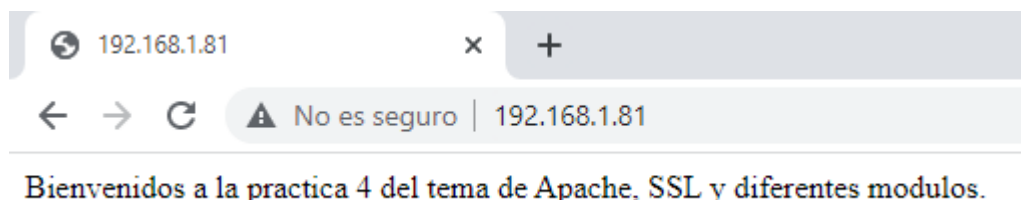
```
GNU nano 4.8                               sitio4ssl.conf
<IfModule mod_ssl.c>
  <VirtualHost _default_:443>

      ServerAdmin webmaster@localhost
      DocumentRoot /var/www

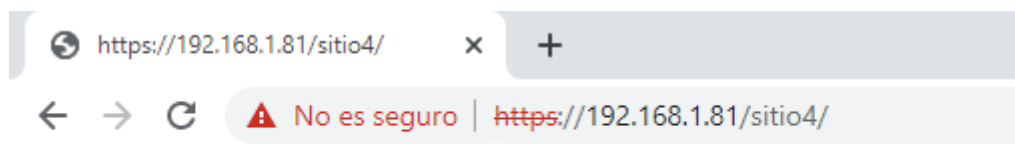
      # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
      # error, crit, alert, emerg.
      # It is also possible to configure the loglevel for particular
      # modules, e.g.
      #LogLevel info ssl:warn

      ErrorLog ${APACHE_LOG_DIR}/error.log
      CustomLog ${APACHE_LOG_DIR}/access.log combined
```

Si yo inserto en el navegador nuestra ip precedida de http:// nos mostrará nuestro sitio4 y su contenido en index.html.



En cambio, si lo hago precedido de https nos mostrará una advertencia. Yo accedo mediante configuración avanzada y nuestro sitio nos será revelado junto con el contenido del archivo index.html.



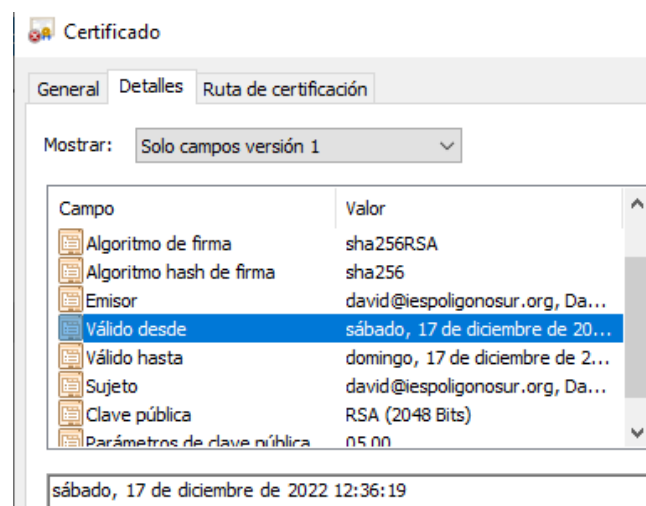
Bienvenidos a la practica 4 del tema de Apache, SSL y diferentes modulos.

3. CERTIFICADO DIGITAL SSL

Al inicio de este documento, ya quedo definida la función de la tecnología SSL. Pues bien, ahora vamos a crear un certificado ssl para así autenticar la identidad de un sitio web, en este caso el sitio4, habilitando una conexión cifrada. Para proceder, tendremos que insertar la siguiente línea de comando donde daremos nombre a nuestra clave privada y al propio certificado.

```
usuario@usuario:/etc/apache2/sites-available$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/davidssl.key -out /etc/ssl/certs/davidssl.crt
[sudo] password for usuario:
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/etc/ssl/private/davidssl.key'
-----
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]:Seville
Locality Name (eg, city) []:Seville
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:David
Email Address []:david@iespoligonosur.org
```

Este par de claves deben insertarse en el archivo de configuración ssl del sitio4. Cuando se introduzcan, solo habrá que reiniciar el servicio apache para aplicar los cambios. Si añado en el navegador nuestra ip precedida de nuevo de https, podremos verificar nuestro certificado.



4. MÓDULO REWRITE Y REDIRECT

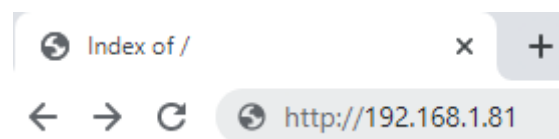
Si necesitamos redirigir un determinado contexto de HTTP a HTTPS podemos hacerlo con **redirect**. Este método es muy sencillo de implantar, puesto que solo será necesario introducir en el archivo de configuración del sitio la directiva que podrá verse en la siguiente imagen.

```
GNU nano 4.8                               sitio4.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

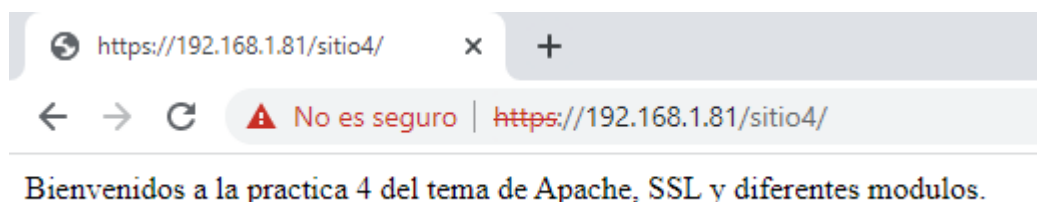
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www

    <Directory "/var/www">
        Redirect / "https://192.168.1.81/"
    </Directory>_
```

Aplicamos los cambios reiniciando el servicio apache y si vamos al navegador e inserto la ip del sitio precedida de http:



Lo que sucederá es que se nos redirigirá a https, al puerto 443.



El módulo **rewrite** da una gran flexibilidad a Apache a la hora de implementar redirecciones que modifiquen el comportamiento de nuestro servidor web: bloquear IPs, forzar tráfico HTTPS para determinados contextos, etc... Sin mayores preámbulos, activo dentro de los módulos disponibles el destinado a rewrite.

```
usuario@usuario:/etc/apache2/mods-available$ sudo a2enmod rewrite
Enabling module rewrite.
To activate the new configuration, you need to run:
systemctl restart apache2
```

Dentro del archivo de configuración del sitio4, implementaremos la directiva encargada de activar el rewrite. Si se observa detenidamente, la directiva cuenta con tres líneas rewrite. La primera de ellas activa rewrite y la segunda establece una condición que se aplicará cada vez que se haga una petición al servidor para el puerto 80. La tercera de las líneas, algo más compleja, se encarga de crear una dirección HTTPS a partir de una consulta HTTP. Entre corchetes vemos REQUEST_URI, que viene a decir que se aplicará a cada index alojados del principal de forma hereditaria.

```
GNU nano 4.8                               sitio4.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www

    <Directory "/var/www">
        RewriteEngine On
        RewriteCond %{SERVER_PORT} 80
        RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
    </Directory>
```

Si volvemos a introducir la ip en el navegador precedida de http, nos redirigirá de forma automática a nuestro puerto 443.



- **Diferencias entre Redirect y Rewrite**

La principal diferencia entre Redirect y Rewrite puede ser descrito sucintamente:

- Rewrite es una operación del lado del servidor.
- Redirect envía una respuesta inmediata al cliente.

En cuanto a rewrite, la solicitud se reescribe, lo cual viene a significar que el resultado de la reescritura es realizado en el lado del servidor. El resultado podría ser una página estática, o un archivo de imagen o una página dinámica (como una página php). Independientemente del resultado de la reescritura de URL, el navegador no es notificado que se ha producido una nueva versión, dentro del servidor web.

En cuanto a redirect, la redirección implica el envío de una respuesta HTTP al cliente, con una nueva dirección URL. El servidor devuelve una respuesta HTTP, con un código de respuesta 301 (Movido permanentemente) o 302 (Movido temporalmente), con una nueva dirección.

5. MÓDULO USERDIR

Userdir es un módulo de apache que hace posible que todos los usuarios con acceso a un servidor tengan una carpeta llamada public_html en la cual puedan alojar sus páginas y archivos. La habilitación es bastante sencilla, pues bastará con activar el módulo userdir. Cuando se active este módulo, reiniciamos el servicio apache para aplicar los cambios.

```
usuario@usuario:/etc/apache2/sites-available$ sudo a2enmod userdir
Enabling module userdir.
To activate the new configuration, you need to run:
systemctl restart apache2
```

Dentro del directorio home de mi usuario, creo una carpeta llamada public_html. Dentro de ella creo un archivo index.html.

```
usuario@usuario:~$ mkdir public_html
```

```
GNU nano 4.8 index.html
Prueba del modulo Userdir by David Segura Tristanco
```

Si nos vamos al navegador e insertamos nuestra ip seguida de una barra transversal, una virgulilla y el usuario, nos mostrara el archivo de dicho usuario.

