ACTIVAR LA SEGURIDAD EN APACHE (HTTPS)

SERVICIOS EN RED

JUAN CARLOS NAVIDAD GARCÍA

Índice:

1. ¿Qué es HTTPS y SSL?: 3
2. ¿Qué es el módulo SSL?: 3
3. Activación del módulo SSL: 3
4. Creación del host virtual con HTTPS: 5
5. Creación del dominio en el servidor DNS: 5
6. Comprobación del host virtual sin HTTPS: 7
7. Creación de las claves SSL: 7
8. Configuración del host virtual con HTTPS: 10
9. Comprobación del host virtual con HTTPS: 11

1. ¿Qué es HTTPS y SSL?:

HTTPS (HyperText Transfer Protocol Secure, protocolo seguro de transferencia de hipertexto) es un protocolo de comunicación de Internet que protege la integridad y la confidencialidad de los datos de los clientes y el sitio web.
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.
Nosotros podremos hacer uso del protocolo **HTTPS** mediante el módulo **SSL**.

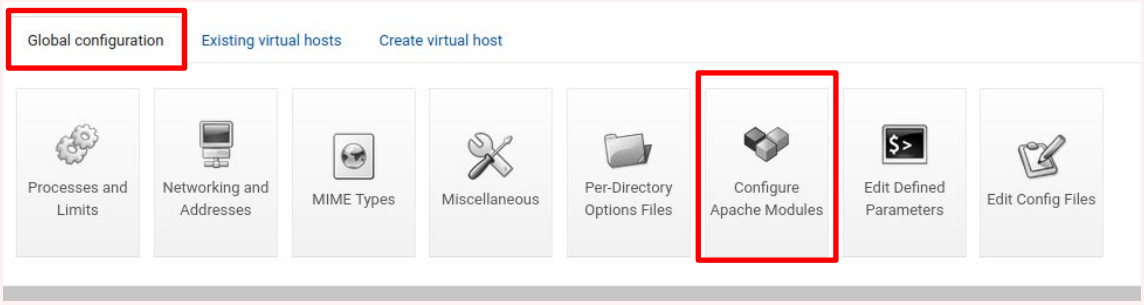
2. ¿Qué es el módulo SSL?:

El módulo **SSL** te da la posibilidad de poder insertar un **certificado SSL**, básicamente, un certificado digital que autentica la identidad de un sitio web y habilita una conexión cifrada, lo que hace que la página sea **HTTPS**.

3. Activación del módulo SSL:

El módulo **SSL** no viene activado por defecto en **Apache2**. Así que, diré dónde y cómo se puede activar, para así comprobar si está activado y o si no lo está, activarlo.

Entonces, para activarlo, nos iremos al apartado de **Configuración global** → **Configuración de Módulos Apache**:



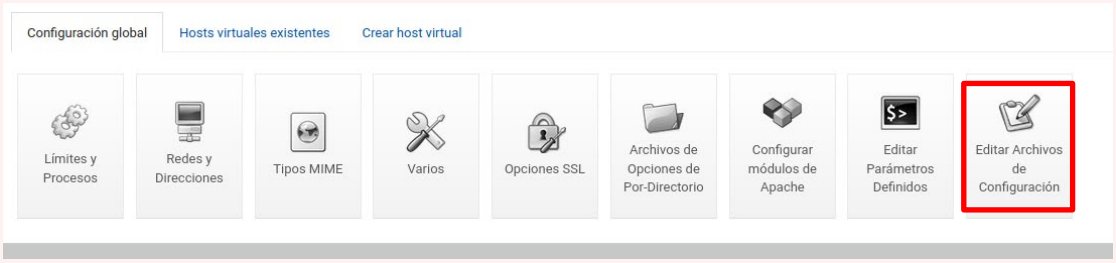
Dentro de los **Módulos de Apache**, buscaremos el módulo **SSL** y comprobaremos si está activado o no, en el caso de que no lo esté, simplemente clicamos sobre el **Checkbox**, del módulo para **activarlo**.

Después de eso, le daremos a **Guardar** y **Aplicar** cambios para que toda la configuración se guarde y se realice sobre el **servidor**.

Podemos ver si el módulo se ha activado correctamente visualizando el archivo de configuración */etc/apache2/ports.conf*, en este archivo debe de aparecer la línea **<IfModule ssl_module>** con el **puerto 443**.
Para visualizar este archivo lo podemos hacer desde la terminal mediante el comando:

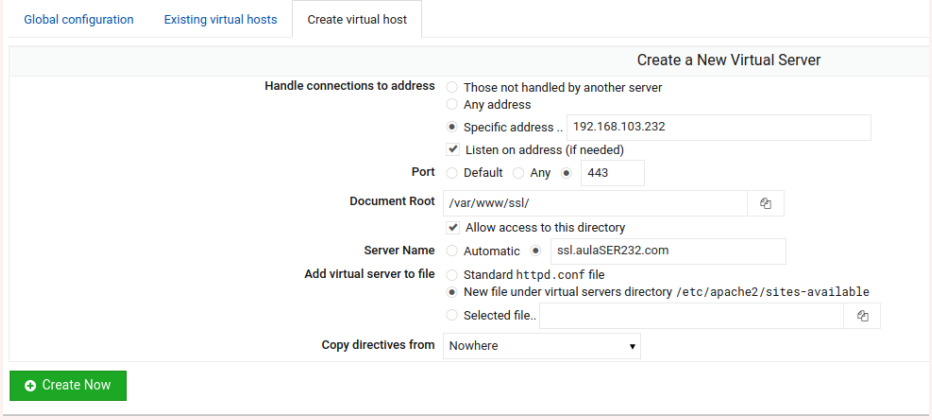
- **sudo nano** */etc/apache2/ports.conf*

También lo podemos visualizar directamente desde **Webmin** desde el apartado de **Configuración global → Editar Archivos de Configuración** y en la lista seleccionaremos el archivo */etc/apache2/ports.conf*:

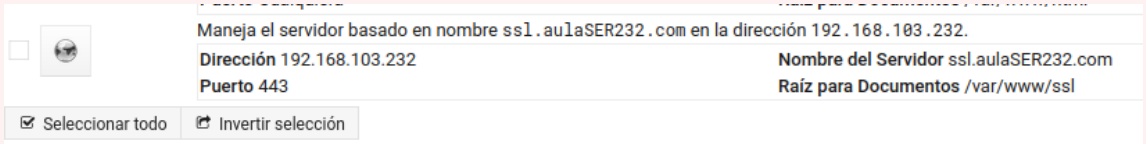


4. Creación del host virtual con HTTPS:

En esta práctica haremos uso de un **host virtual** para que la página que tenga sea la única con **HTTPS** y no afecte a las demás.
Como ya hemos hecho en prácticas anteriores, añadiremos el **host virtual** desde el apartado **Crear un nuevo host virtual** y el único cambio respecto a los anteriores creados, es que el puerto será el **443**.

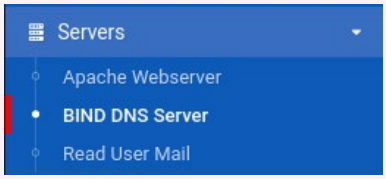


Se nos creará el **host virtual**:

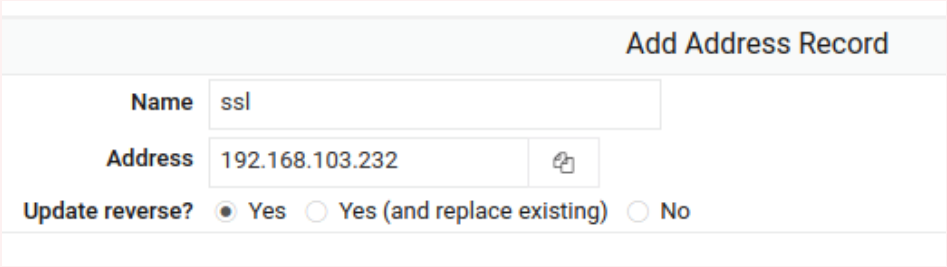


5. Creación del dominio en el servidor DNS:

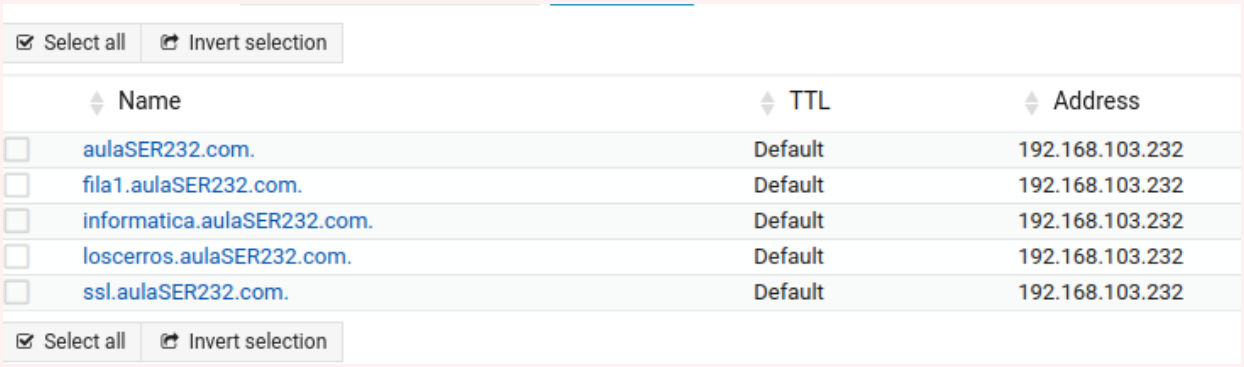
Añadir un dominio a nuestro **servidor DNS** es una de las cosas que ya hemos hecho, igualmente, para añadirlo, primero nos debemos de ir al **servidor BIND DNS** desde el panel de navegación a la izquierda de **Webmin**:



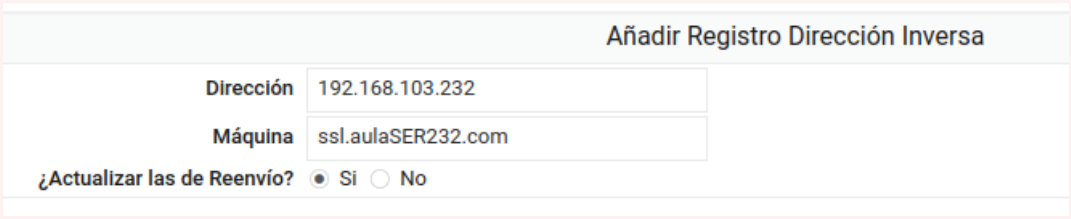
Nos iremos a nuestra zona de **búsqueda directa** y añadiremos el **dominio** desde **Direcciones**:



Tendremos todos los **dominios** creados anteriormente para los demás **hosts virtuales** más el nuevo que hemos creado dedicado para **HTTPS**:



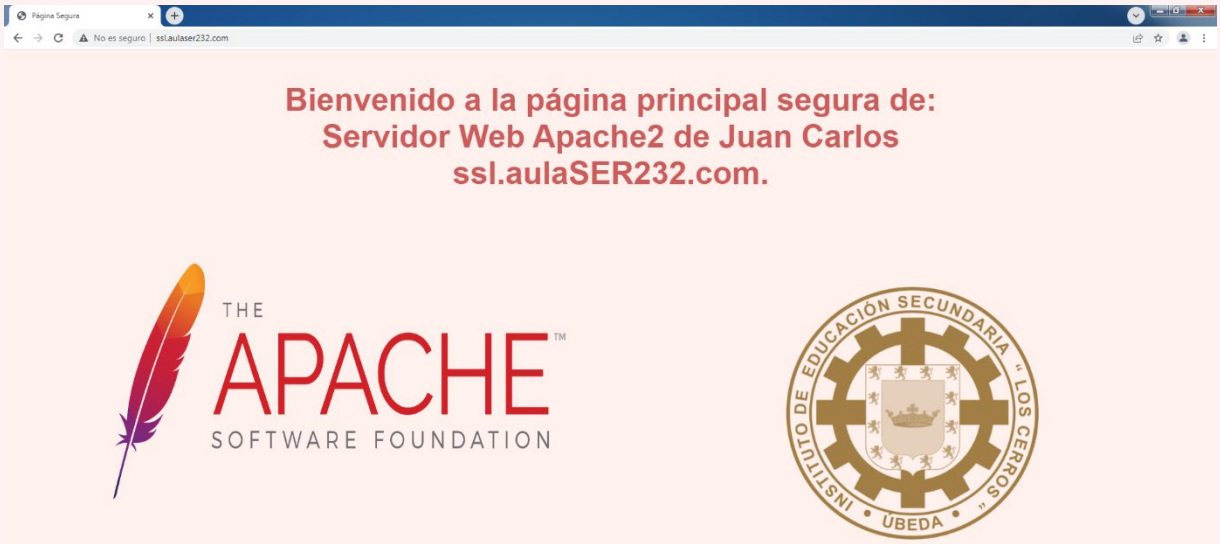
Como tenemos más de un dominio asignado a una misma **dirección IP** este no se añadirá automáticamente a la zona de **búsqueda inversa**, así que lo añadiremos de la misma manera, nos iremos a la zona y pulsaremos sobre **Direcciones inversas**:



Tendremos todos los **dominios** creados anteriormente para los demás **hosts virtuales** más el nuevo que hemos creado dedicado para **HTTPS**:

6. Comprobación del host virtual sin HTTPS:

Antes de nada, vamos a comprobar si el **host virtual** y el **dominio** que le hemos asignado funciona correctamente, a parte de la página web que le hemos diseñado.
Si accedemos desde otro equipo o máquina con nuestra **DNS** configurada e introducimos el **dominio** del **host virtual** en el navegador, debería de funcionar la página creada:



7. Creación de las claves SSL:

Sabiendo que todo funciona bien, vamos crear las **claves SSL**. La orden para crear el **certificado SSL** es **apache2-ssl-certificate**, pero no va incluida en **Linux**. Se ha de descargar e instalar el archivo **Apache2-ssl.tar.gz.** Para ellos, ejecuta la siguiente orden desde un **terminal**:

- **sudo wget** *http://librarian.launchpad.net/7477840/apache2-ssl.tar.gz*

```
jnav@jnav-vb:~$ sudo wget http://librarian.launchpad.net/7477840/apache2-ssl.tar.gz
--2022-02-10 22:49:41--  http://librarian.launchpad.net/7477840/apache2-ssl.tar.gz
Resolviendo librarian.launchpad.net (librarian.launchpad.net)... 91.189.89.225, 91.18
9.89.224, 2001:67c:1560:8003::8004, ...
Conectando con librarian.launchpad.net (librarian.launchpad.net)[91.189.89.225]:80...
 conectado.
Petición HTTP enviada, esperando respuesta... 301 Moved Permanently
Ubicación: https://launchpadlibrarian.net/7477840/apache2-ssl.tar.gz [siguiente]
--2022-02-10 22:49:41--  https://launchpadlibrarian.net/7477840/apache2-ssl.tar.gz
Resolviendo launchpadlibrarian.net (launchpadlibrarian.net)... 91.189.89.229, 91.189.
89.228, 2001:67c:1560:8003::8008, ...
Conectando con launchpadlibrarian.net (launchpadlibrarian.net)[91.189.89.229]:443...
conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 964 [application/x-tar]
Guardando como: “apache2-ssl.tar.gz”

apache2-ssl.tar.gz  100%[==================>]     964  --.-KB/s    en 0s

2022-02-10 22:49:42 (37,7 MB/s) - “apache2-ssl.tar.gz” guardado [964/964]

jnav@jnav-vb:~$
```

Una vez ejecutado el comando anterior, se nos descargará el archivo **Apache-ssl.tar.gz** el cual tendremos que descomprimir:

```
jnav@jnav-vb:~$ ls
apache2-ssl.tar.gz
```

Para descomprimirlo utilizaremos el comando **sudo tar -xvf apache2-ssl.tar.gz**, este archivo contendrá los archivos **apache2-ssl-certificate** y **ssleay.cnf**:

```
jnav@jnav-vb:~$ tar -xvf apache2-ssl.tar.gz
apache2-ssl-certificate
ssleay.cnf
```

Lo siguiente que haremos será copiar el archivo **apache2-ssl-certificate** al directorio */usr/bin/*

```
jnav@jnav-vb:~$ sudo cp apache2-ssl-certificate /usr/bin
jnav@jnav-vb:~$ ls /usr/bin/ | egrep "apache2"
apache2-ssl-certificate
jnav@jnav-vb:~$
```

El otro archivo que queda, lo mandaremos al directorio */usr/share/apache2/*:

```
jnav@jnavrog:~$ sudo cp ssleay.cnf  /usr/share/apache2/
```

Ahora, crearemos los directorios donde se van a guardar las **claves SSL**.
El primero en crearse será el */etc/apache2/ssl/*, al cual también le daremos **permisos de ejecución**:

```
jnav@jnav-vb:~$ sudo mkdir /etc/apache2/ssl/
jnav@jnav-vb:~$
```

```
jnav@jnav-vb:~$ sudo chmod +x /etc/apache2/ssl/
```

Ahora, dentro del directorio */etc/apache2/ssl/*, crearemos la carpeta **miCA** y en su interior otra carpeta llamada **private**:

```
jnav@jnav-vb:/etc/apache2/ssl$ sudo mkdir miCA
jnav@jnav-vb:/etc/apache2/ssl$ sudo mkdir miCA/private
```

Cuando tengamos todos los directorios creados, vamos a generar las **claves** con el comando:

- **sudo apache2-ssl-certificate**

Nos pedirá diferente información para la creación de los **certificados**, tenemos la opción de darle a **Enter** todo el rato y saltarla, pero se recomienda ponerla:

```
jnav@jnav-vb:~$ sudo apache2-ssl-certificate

creating selfsigned certificate
replace it with one signed by a certification authority (CA)

enter your ServerName at the Common Name prompt
```

```
-----
Country Name (2 letter code) [GB]:ES
State or Province Name (full name) [Some-State]:Jaén
Locality Name (eg, city) []:Úbeda
Organization Name (eg, company; recommended) []:LosCerros
Organizational Unit Name (eg, section) []:Educación
server name (eg. ssl.domain.tld; required!!!) []:aulaSER232.com
Email Address []:navidadgarcia.juancarlos@loscerros.org
jnav@jnav-vb:~$
```

Las claves se generarán en el directorio creado anteriormente, */etc/apache2/ssl/*.
Accederemos al directorio */etc/apache2/ssl/* y copiaremos la clave **apache.pem** a la carpeta **miCA** con el nombre **cacert.pem** y también lo copiaremos al directorio */miCA/private/* bajo el nombre de **cakey.pem**:

```
jnav@jnav-vb:/etc/apache2/ssl$ sudo cp apache.pem miCA/private/cakey.pem
jnav@jnav-vb:/etc/apache2/ssl$ sudo cp apache.pem miCA/cacert.pem
jnav@jnav-vb:/etc/apache2/ssl$
```

También, crearemos el archivo de texto **serial** con una única línea de texto que sería **01**:

```
jnav@jnav-vb:/etc/apache2/ssl$ sudo nano miCA/serial
jnav@jnav-vb:/etc/apache2/ssl$
```

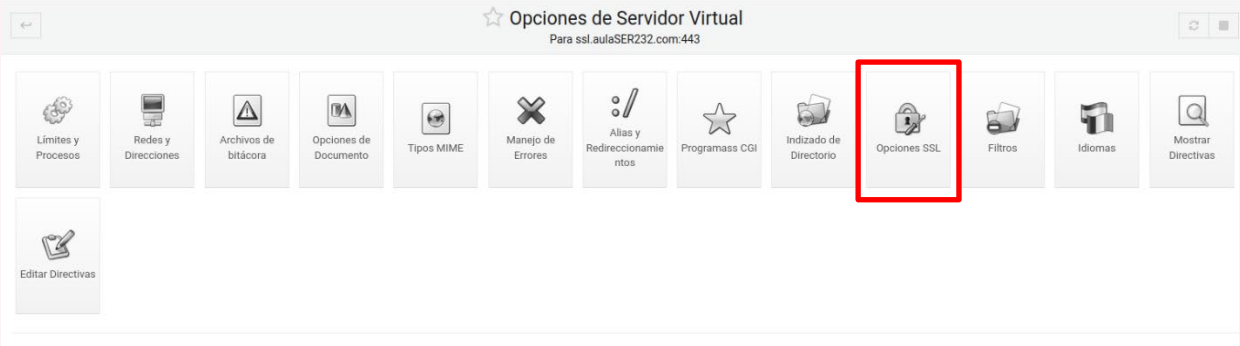
```
  GNU nano 2.9.3                          miCA/serial

01
```

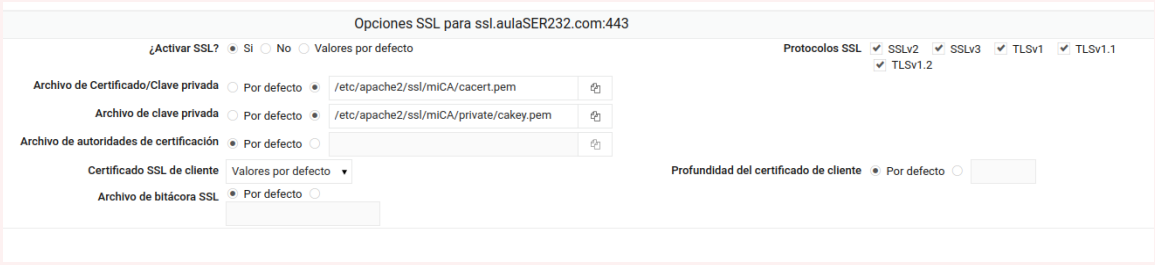
Con todo esto realizado, las **claves** ya estarían generadas y guardadas en sus correspondientes directorios para poder insertarlas en el **host virtual** y que funcione con el protocolo **HTTPS.**

8. Configuración del host virtual con HTTPS:

Nos iremos al **host virtual** y al tener configurado el **host virtual** con el **puerto 443** que corresponde a **SSL**, nos aparecerá un apartado que es **Opciones SSL**, tendremos que acceder a él:



Dentro de las opciones **SSL**, debemos de activar **SSL** y insertarle los dos archivos de **claves**, primero el */etc/apache2/ssl/miCA/cacert.pem* y segundo el */etc/apache2/ssl/miCA/private/cakey.pem*:

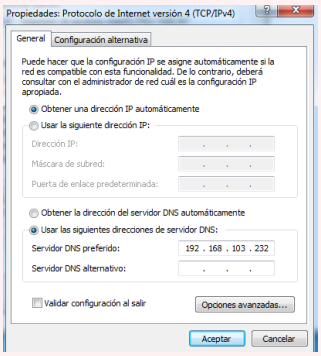


Guardaremos y le daremos al botón de **aplicar cambios**:

Si todo lo hemos hecho bien, no nos debería de dar ningún error, pero sería común que no los diese.
Así que, si todo va, probaremos a acceder desde nuestra máquina configurada con nuestro **servidor DNS**.

9. Comprobación del host virtual con HTTPS:

Una vez realizados todos los pasos anteriores, nuestra **página web** ya estaría protegida con el protocolo HTTPS, por lo tanto, el módulo **SSL** ya estaría configurado. Solo quedaría comprobar que funciona todo correctamente.
Para eso nos iremos a una **máquina virtual** u otro **equipo en nuestra propia red**, asignarle nuestra **DNS** y hacer las comprobaciones correspondientes.



Si tenemos bien configurado el **equipo cliente**, nos iremos a un **navegador** y pondremos nuestro **dominio,** pero con *https://* detrás, poniéndolo se nos redireccionará automáticamente a nuestra página **https**, sino podemos poner directamente el **dominio** creado delante de *https://*.