



# Hardening Apache

Ricardo Sorin Almajan

## Índice

Configuración /etc/apache2/apache2.conf .....	2
Verificando el usuario encargado de ejecutar apache.....	4
Configurando Headers de respuesta.....	5
TLS .....	6
Configurando TLS. ....	6
Configurando Apache.....	7
Configurando Apache.....	7
Configurar archivo de virtual host.....	7
Redireccionando HTTP hacia HTTPS.....	8
Aplicar configuración. ....	8
Verificando configuración. ....	11

## Configuración /etc/apache2/apache2.conf

Ejemplo:

L	Configuración /etc/apache2/apache2.conf
1	# This is the main Apache server configuration file.
2	# The directory where shm and other runtime files will be stored.
3	DefaultRuntimeDir \${APACHE_RUN_DIR}
4	# PidFile: The file in which the server should record its process
5	PidFile \${APACHE_PID_FILE}
6	#SP-Server Configuration
7	Timeout 30
8	# KeepAlive: Whether or not to allow persistent
9	KeepAlive On
10	# MaxKeepAliveRequests: The maximum number of requests to allow
11	during a persistent connection
12	MaxKeepAliveRequests 100
13	#SP-Server Configuration
14	KeepAliveTimeout 3
15	# These need to be set in /etc/apache2/envvars
16	User \${APACHE_RUN_USER}
17	Group \${APACHE_RUN_GROUP}

```
#<Directory /srv/>
#     Options Indexes FollowSymLinks
#     AllowOverride None
#     Require all granted
#</Directory>
<Directory /var/www/www.motos.com/>
    Options None
    AllowOverride None
    Require all granted
    <LimitExcept POST GET HEAD >
    Deny from all
    </LimitExcept>
</Directory>
```

Hay que ir buscando las líneas y dejarlas igual para después guardar el archivo

```

18 # HostnameLookups: Log the names of clients or just their IP addresses
19 HostnameLookups Off
20 # ErrorLog: The location of the error log file.
21 ErrorLog ${APACHE_LOG_DIR}/error.log
22 # LogLevel: Control the severity of messages logged to the error_log.
23 LogLevel warn
24 # Include module configuration:
25 IncludeOptional mods-enabled/*.load
26 IncludeOptional mods-enabled/*.conf
27 # Include list of ports to listen on
28 Include ports.conf
29 # Sets the default security model of the Apache2 HTTPD server.
30 <Directory /var/www/www.sp/>
31     Options None
32     AllowOverride None
33     Require all granted
34     <LimitExcept POST GET HEAD >
35         Deny from all
36     </LimitExcept>
37 </Directory>
38 # AccessFileName.
39 AccessFileName .htaccess
40 <FilesMatch "\.ht">
41     Require all denied
42 </FilesMatch>
43 # Include generic snippets of statements
44 IncludeOptional conf-enabled/*.conf
45 # Include the virtual host configurations:
46 IncludeOptional sites-enabled/*.conf
47 #SP-Server Configuration
48 ServerTokens Prod
49 #SP: Evitando que el servidor exponga información
50 ServerSignature Off

```

*Ilustración 2-1. fichero de configuración /etc/apache2/apache2.conf*

Con esto lo que se busca es reducir el numero de ataques posibles por ejemplo con el parámetro timeout, lo que hacemos es evitar ataques de denegación de servicio (DOS) y ataques de slowloris pero teniendo en cuenta también el rendimiento del servidor.

El parámetro AllowOverride tal como lo hemos establecido hace que los usuarios no puedan configurar ficheros .htaccess que pueden invalidar características de seguridad que se hayan establecido.

Una vez dejado todo habrá que reiniciar apache:

Ahora al poner localhost/ no nos debería de salir el index

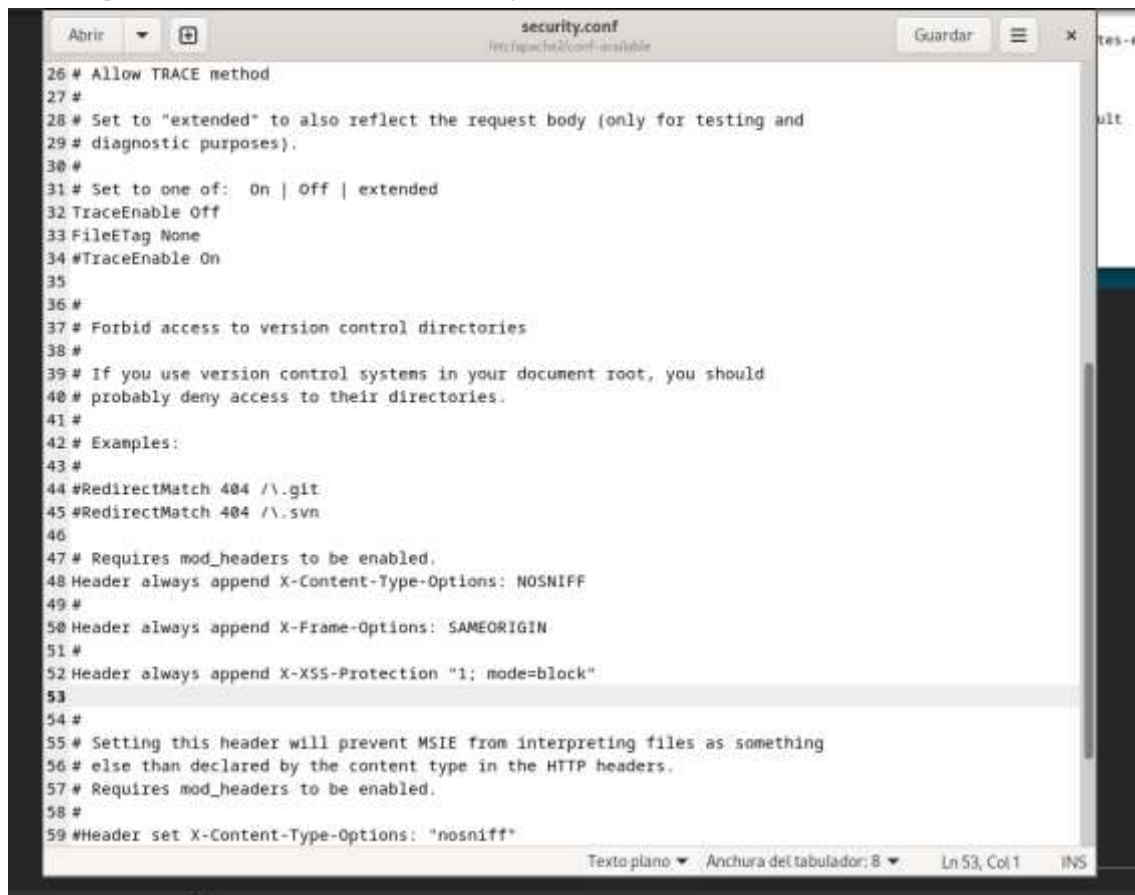


## Verificando el usuario encargado de ejecutar apache

```
x@debian:/etc/apache2$ sudo apachectl -S
VirtualHost configuration:
*:80               is a NameVirtualHost
                    default server debian.com (/etc/apache2/sites-enabled/000-default.conf:
1)
                    port 80 namevhost debian.com (/etc/apache2/sites-enabled/000-default.co
nf:1)
                    port 80 namevhost motos.com (/etc/apache2/sites-enabled/motos.conf:1)
ServerRoot: "/etc/apache2"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/var/log/apache2/error.log"
Mutex default: dir="/var/run/apache2/" mechanism=default
Mutex watchdog-callback: using_defaults
PidFile: "/var/run/apache2/apache2.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="www-data" id=33
Group: name="www-data" id=33
x@debian:/etc/apache2$
```

Por norma general el usuario que usa apache no debe de tener ningún tipo de privilegio, en caso de que se encuentra uno es muy recomendable cambiarlo.

## Configurando Headers de respuesta



```
26 # Allow TRACE method
27 #
28 # Set to "extended" to also reflect the request body (only for testing and
29 # diagnostic purposes).
30 #
31 # Set to one of: On | Off | extended
32 TraceEnable Off
33 FileETag None
34 #TraceEnable On
35
36 #
37 # Forbid access to version control directories
38 #
39 # If you use version control systems in your document root, you should
40 # probably deny access to their directories.
41 #
42 # Examples:
43 #
44 #RedirectMatch 404 /\.git
45 #RedirectMatch 404 /\.svn
46
47 # Requires mod_headers to be enabled.
48 Header always append X-Content-Type-Options: NOSNIFF
49 #
50 Header always append X-Frame-Options: SAMEORIGIN
51 #
52 Header always append X-XSS-Protection "1; mode=block"
53
54 #
55 # Setting this header will prevent MSIE from interpreting files as something
56 # else than declared by the content type in the HTTP headers.
57 # Requires mod_headers to be enabled.
58 #
59 #Header set X-Content-Type-Options: "nosniff"
```

En la foto anterior se puede observar la configuración de los Headers que se encuentran en el archivo `/etc/apache2/conf-available/security.conf`

Lo que hay que implementar:

L	<code>/etc/apache2/conf-available/security.conf</code>
1	<code># Requires mod_headers to be enabled.</code>
2	<code>Header always append X-Content-Type-Options: NOSNIFF</code>
3	<code>#</code>
4	<code>Header always append X-Frame-Options: SAMEORIGIN</code>
5	<code>#</code>
6	<code>Header always append X-XSS-Protection "1; mode=block"</code>

Estos Headers ayudan a mejorar la seguridad, `X-Content-Type-Options` previene que los navegadores cambien el tipo de archivo de un elemento web, evitando vulnerabilidades de sniffing MIME. `X-Frame-Options`, al configurarse con el valor `"SAMEORIGIN"`, impide que una página se cargue en un `iframe` de un origen distinto, protegiendo contra ataques de clickjacking.

De la línea 46-52 hay que dejarlo igual en el fichero.

```
r@debian:/etc/apache2/conf-available$ sudo a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
    systemctl restart apache2
r@debian:/etc/apache2/conf-available$ sudo sytemctl restar apache2
```

```
* sudo systemctl restart apache2
```

Una vez modificado el archivo habrá que reiniciar apache para que e aplique la configuración.

## TLS

TLS (Transport Layer Security) es la versión más actualizada y segura de SSL, utilizada para establecer una comunicación segura entre un servidor web y un cliente, evitando que un atacante pueda leer o modificar los datos transferidos.

## Configurando TLS.

```
root@kali:~/jetc/papache# sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout jetc/ssl/private/sp-server.key -out jetc/ssl/certs/sp-server.crt
```

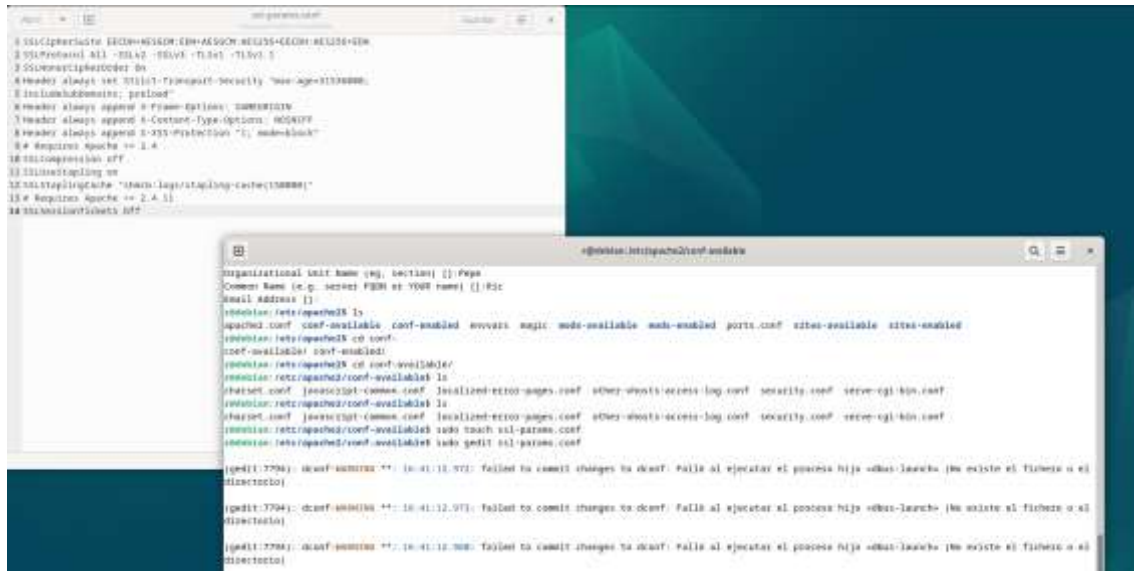
En ese comando lo que estamos haciendo es crearla clave privada y el certificado publico haciendo uso de openssl, una vez realizado nos pedirá cierta información del servidor.

```
# sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/sp-server.key -out /etc/ssl/certs/sp-server.crt
```

## Configurando Apache

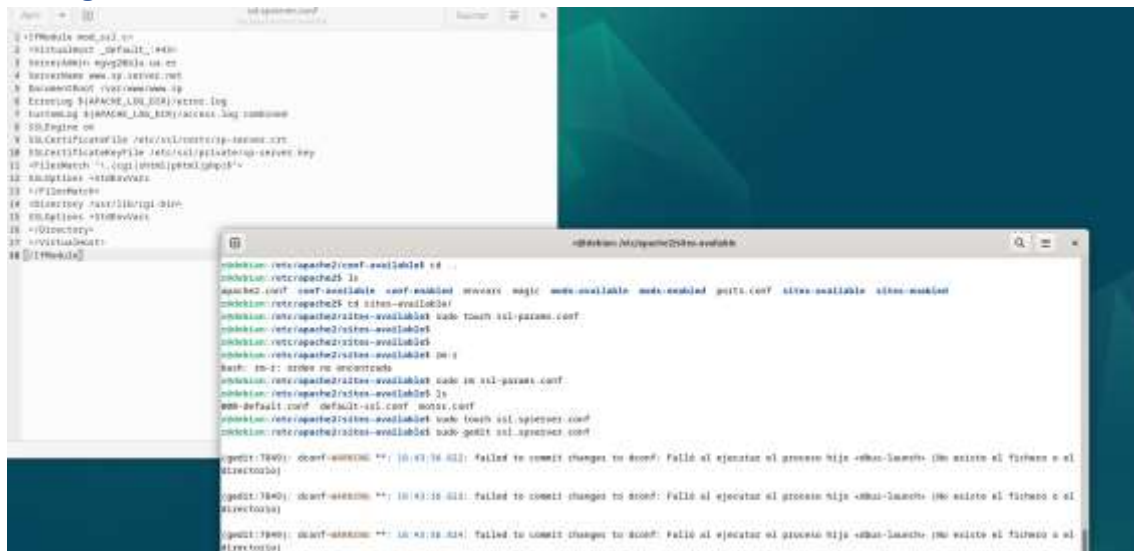
Se crea el fichero `/etc/apache2/conf-available/ssl-params.conf`, que contiene información de configuración que se cargará en Apache. Este contendrá información de este y la configuración de SSL

## Configurando Apache



En este fichero se implementa un nuevo encabezado una vez ya implementadas conexiones seguras SSL, este encabezado establece un mecanismo que redirige automáticamente cualquier intento de conexión insegura (HTTP) a una conexión segura (HTTPS).

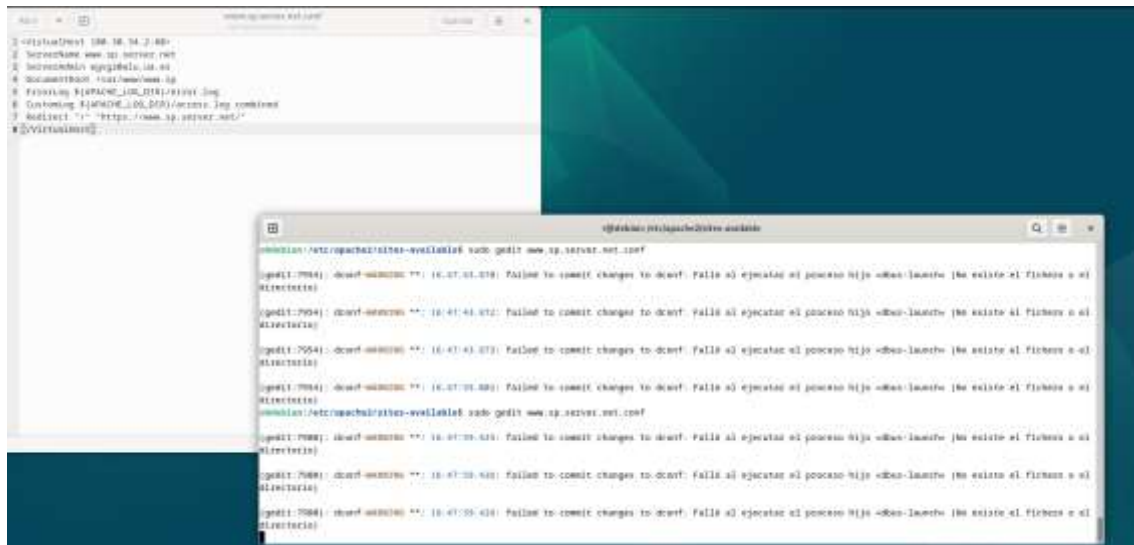
### Configurar archivo de virtual host.



Se crea el fichero `/etc/apache2/sites-available/ssl.spserver.conf`. en este fichero se configurará la información de un virtual host típico



## Redireccionando HTTP hacia HTTPS.



Configuraremos el fichero /etc/apache2/sites-available/www.sp.server.net.conf añadiendo un redirect al sitio que se desea.

### Aplicar configuración.

```
r@debian:/etc/apache2/sites-available$ a2enmod ssl
bash: a2enmod: orden no encontrada
r@debian:/etc/apache2/sites-available$ sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
r@debian:/etc/apache2/sites-available$ sudo a2enconf ssl-params
Enabling conf ssl-params.
To activate the new configuration, you need to run:
    systemctl reload apache2
r@debian:/etc/apache2/sites-available$ sudo a2ensite ssl.spserver.conf
Enabling site ssl.spserver.
To activate the new configuration, you need to run:
    systemctl reload apache2
r@debian:/etc/apache2/sites-available$ sudo systemctl reload apache2
```

Para aplicar la configuración hay que habilitar unos módulos, ssl, ssl-params.conf y el del virtual host

Si da este error:

```
r@debian:/etc/apache2/sites-available$ sudo systemctl reload apache2
```

Job for apache2.service failed.

See "systemctl status apache2.service" and "journalctl -xeu apache2.service" for details.

Hacer systemctl status apache2.service

Y en la información se puede observar donde da el fallo, en este caso es del arco ssl-params.conf

```
ene 20 16:55:07 debian apachectl[8088]: Action 'graceful' failed.
ene 20 16:55:07 debian apachectl[8088]: The Apache error log may have more information.
r@debian:/etc/apache2/sites-available$ 
r@debian:/etc/apache2/sites-available$ ^C
r@debian:/etc/apache2/sites-available$ sudo nano /etc/apache2/conf-enabled/ssl-params.conf
r@debian:/etc/apache2/sites-available$ sudo systemctl reload apache2
r@debian:/etc/apache2/sites-available$
```

El archivo de debería de ver así

SSLCipherSuite ECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH

SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1

SSLHonorCipherOrder On

Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains; preload"

Header always append X-Frame-Options: SAMEORIGIN

Header always append X-Content-Type-Options: NOSNIFF

Header always append X-XSS-Protection "1; mode=block"

# Requires Apache >= 2.4

SSLCompression off

SSLUseStapling on

SSLStaplingCache "shmcb:logs/stapling-cache(150000)"

# Requires Apache >= 2.4.11

SSLSessionTickets Off

El fallo estaba en que al poner la configuración hubo un error de sintaxis.

```
r@debian:/etc/apache2/sites-available$ sudo nano /etc/apache2/conf-enabled/ssl-params.conf
r@debian:/etc/apache2/sites-available$ sudo systemctl reload apache2
r@debian:/etc/apache2/sites-available$ sudo systemctl status apache2.service
* apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-01-20 16:33:51 CET; 37min ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 7668 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
  Process: 8141 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
 Main PID: 7674 (apache2)
    Tasks: 55 (limit: 2252)
   Memory: 11.2M
      CPU: 772ms
   CGroup: /system.slice/apache2.service
           └─7674 /usr/sbin/apache2 -k start
```

```
ene 20 16:55:07 debian apache2ctl[8088]: Action 'graceful' failed.
ene 20 16:55:07 debian apache2ctl[8088]: The Apache error log may have more information.
ene 20 16:55:07 debian systemd[1]: apache2.service: Control process exited, code=exited, status=1/FAILURE
ene 20 16:55:07 debian systemd[1]: Reload failed for apache2.service - The Apache HTTP Server.
ene 20 17:10:18 debian systemd[1]: Reloading apache2.service - The Apache HTTP Server...
r@debian:/etc/apache2/sites-available$ sudo nano /etc/apache2/conf-enabled/ssl-params.conf
r@debian:/etc/apache2/sites-available$ sudo a2enmod headers
Module headers already enabled
r@debian:/etc/apache2/sites-available$ sudo systemctl reload apache2
r@debian:/etc/apache2/sites-available$ sudo apache2ctl configtest
AH00112: Warning: DocumentRoot [/var/www/www.sp] does not exist
Syntax OK
r@debian:/etc/apache2/sites-available$ sudo systemctl reload apache2
sudo systemctl status apache2
* apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-01-20 16:33:51 CET; 42min ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 7668 ExecStart=/usr/sbin/apache2l start (code=exited, status=0/SUCCESS)
  Process: 8297 ExecReload=/usr/sbin/apache2l graceful (code=exited, status=0/SUCCESS)
    Main PID: 7674 (apache2)
       Tasks: 55 (limit: 2252)
      Memory: 11.3M
         CPU: 1.079s
```

```
ene 20 17:15:44 debian apache2ctl[8238]: AH00112: Warning: DocumentRoot [/var/www/www.sp] does not exist
ene 20 17:15:44 debian systemd[1]: Reloading apache2.service - The Apache HTTP Server.
ene 20 17:16:09 debian systemd[1]: Reloading apache2.service - The Apache HTTP Server...
r@debian:/etc/apache2/sites-available$ ls -l /etc/apache2/conf-enabled/ssl-params.conf
lrwxrwxrwx 1 root root 39 ene 20 16:49 /etc/apache2/conf-enabled/ssl-params.conf -> ../conf-enabled/ssl-params.conf
r@debian:/etc/apache2/sites-available$ sudo chmod 644 /etc/apache2/conf-enabled/ssl-params.conf
r@debian:/etc/apache2/sites-available$ sudo tail -n 50 /var/log/apache2/error.log
[Mon Jan 20 16:43:16.660588 2025] [mpm_event:notice] [pid 825:tid 825] AH00488: Apache/2.4.62 (Debian) configured -- resuming normal operations
[Mon Jan 20 16:43:16.660606 2025] [core:notice] [pid 825:tid 825] AH00894: Command line: '/usr/sbin/apache2'
[Mon Jan 20 16:20:54.884529 2025] [mpm_event:notice] [pid 825:tid 825] AH00492: caught SIGKILL, shutting down gracefully
[Mon Jan 20 16:20:58.928881 2025] [mpm_event:notice] [pid 7405:tid 7405] AH00489: Apache/2.4.62 (Debian) configured -- resuming normal operations
[Mon Jan 20 16:20:58.929251 2025] [core:notice] [pid 7405:tid 7405] AH00894: Command line: '/usr/sbin/apache2'
[Mon Jan 20 16:33:51.758373 2025] [mpm_event:notice] [pid 7405:tid 7405] AH00492: caught SIGKILL, shutting down gracefully
[Mon Jan 20 16:33:51.862040 2025] [mpm_event:notice] [pid 7674:tid 7674] AH00489: Apache/2.4.62 (Debian) configured -- resuming normal operations
[Mon Jan 20 16:33:51.862040 2025] [core:notice] [pid 7674:tid 7674] AH00894: Command line: '/usr/sbin/apache2'
[Mon Jan 20 17:10:18.624830 2025] [mpm_event:notice] [pid 7674:tid 7674] AH00493: SIGUSR1 received. Doing graceful restart
AH00112: Warning: DocumentRoot [/var/www/www.sp] does not exist
[Mon Jan 20 17:10:18.650401 2025] [ssl:warn] [pid 7674:tid 7674] AH01906: www.sp.server.net:443:0 server certificate is a CA certificate (BasicConstraints:
CA == TRUE)
[Mon Jan 20 17:10:18.650402 2025] [ssl:warn] [pid 7674:tid 7674] AH01909: www.sp.server.net:443:0 server certificate does NOT include an ID which matches th
e server name
[Mon Jan 20 17:10:18.650409 2025] [ssl:error] [pid 7674:tid 7674] AH02217: ssl_stapling_init_cert: can't retrieve issuer certificate! (subject: CN=Ric,OU=Pe
pe,O=Orange,L=Zaragoza,ST=Aragon,C=ES / issuer: CN=Ric,OU=Pepe,O=Orange,L=Zaragoza,ST=Aragon,C=ES / serial: 93678527FF1F5E801607AA1C7BB8546512067 / notbef
```

```
* apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-01-20 16:33:51 CET; 43min ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 7668 ExecStart=/usr/sbin/apache2l start (code=exited, status=0/SUCCESS)
  Process: 8273 ExecReload=/usr/sbin/apache2l graceful (code=exited, status=0/SUCCESS)
    Main PID: 7674 (apache2)
       Tasks: 55 (limit: 2252)
      Memory: 19.3M
         CPU: 1.216s
    CGroup: /system.slice/apache2.service
            └─OCF4 /usr/sbin/apache2 -k start
              └─HTTP /usr/sbin/apache2 -k start
                └─HTTP /usr/sbin/apache2 -k start

ene 20 17:10:18 debian systemd[1]: Reloading apache2.service - The Apache HTTP Server.
ene 20 17:15:44 debian systemd[1]: Reloading apache2.service - The Apache HTTP Server...
ene 20 17:15:44 debian apache2ctl[8238]: AH00112: Warning: DocumentRoot [/var/www/www.sp] does not exist
ene 20 17:15:44 debian systemd[1]: Reloading apache2.service - The Apache HTTP Server.
ene 20 17:16:09 debian systemd[1]: Reloading apache2.service - The Apache HTTP Server...
ene 20 17:16:09 debian apache2ctl[8308]: AH00112: Warning: DocumentRoot [/var/www/www.sp] does not exist
ene 20 17:16:09 debian systemd[1]: Reloading apache2.service - The Apache HTTP Server.
ene 20 17:16:56 debian systemd[1]: Reloading apache2.service - The Apache HTTP Server...
r@debian:/etc/apache2/sites-available$
```

En la foto adjuntada se puede ver tanto el error y la solución de cómo debería verse si todo va bien.

## Verificando configuración.

Al buscar la página nos debería de salir que no es segura ya que estamos utilizando certificados auto firmados.

