

## **CURSO DE ESPECIALIZACIÓN DE CIBERSEGURIDAD**

**IESEICaminàs.**

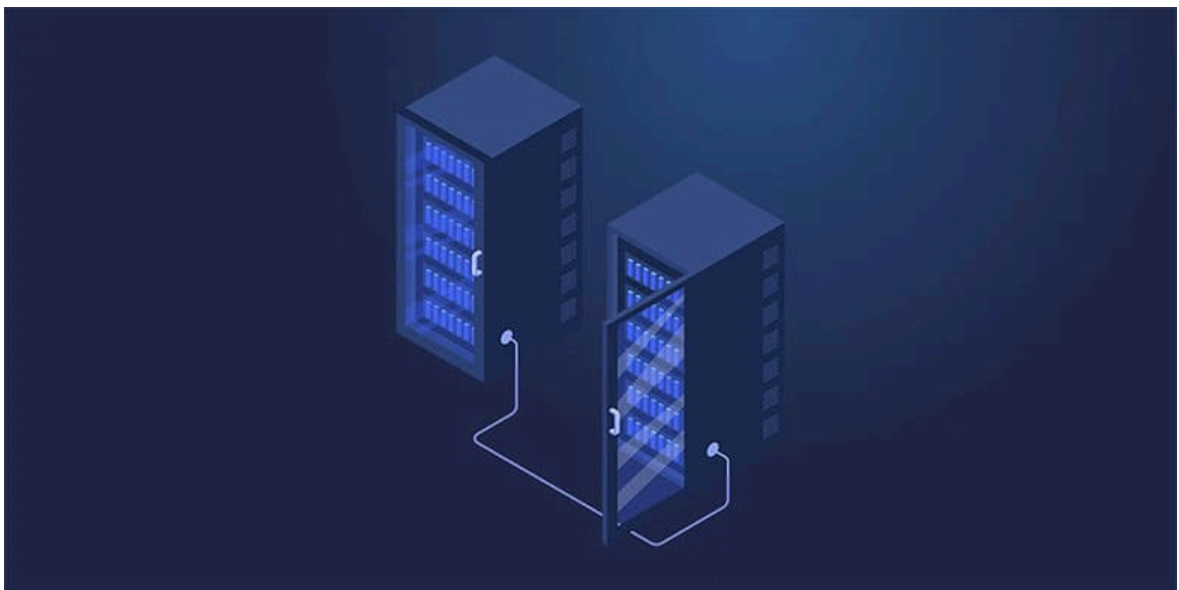
**Módulo: Puesta en producción segura.**

**Título: Server hardening**

**Autor: Jaume Tellols Monfort.**

**Profesor: Pau Conejero Alberola.**

**Fecha: 02/03/2025**



## Objetivos

---

- Implementar medidas de **server hardening** en Apache para reducir vulnerabilidades y mejorar la seguridad del servidor.
- Configurar **Mod Security** y aplicar las reglas OWASP para proteger el servidor contra ataques como **SQL Injection, XSS y RCE**.
- Instalar y configurar **modevasive** para mitigar ataques de **denegación de servicio (DoS)**.
- Implementar un **certificado SSL auto-firmado** en Apache para asegurar la comunicación cifrada entre clientes y servidores.
- Realizar pruebas y validaciones para comprobar la efectividad de las configuraciones de seguridad aplicadas.

## Resumen

---

En esta práctica, hemos aplicado diversas configuraciones de seguridad en un servidor Apache para reforzar su protección contra ataques.

Inicialmente, realizamos server hardening, deshabilitando módulos innecesarios y configurando cabeceras HTTP para mejorar la seguridad. Luego, instalamos y configuramos ModSecurity, integrando las reglas OWASP para detectar y bloquear intentos de ataque. También implementamos mod\_evasive para mitigar ataques DDoS, estableciendo umbrales de solicitudes y realizando pruebas de carga para validar su funcionamiento.

Por último, instalamos un certificado SSL auto-firmado en Apache para habilitar conexiones seguras mediante HTTPS. Configuramos el servidor para utilizar el certificado y realizamos pruebas de acceso, verificando que la conexión estuviera cifrada y funcional. Para asegurar que todas las conexiones fueran seguras, configuramos una redirección automática de HTTP a HTTPS.

## Índice de Contenidos

---

<b>1. Server Hardening.....</b>	<b>3</b>
1.1 Configuración de Seguridad en Apache.....	4
1.2 Implementación de ModSecurity.....	5
1.3 Implementación de Reglas OWASP en ModSecurity.....	6
1.4 Protección Contra Ataques DoS con mod_evasive.....	8
1.5 Instalar un certificado digital en el servidor Apache.....	10
1.6 Apache hardening best practices.....	14
1.6.1 Remove Server Version Banner.....	14
1.6.2 Disable directory browser listing.....	15
1.6.3 Etag.....	15
1.6.4 Run Apache from a non-privileged account.....	16
1.6.5 System Settings Protection.....	17
<b>2. Webgrafía.....</b>	<b>17</b>
<b>3. Conclusión.....</b>	<b>18</b>

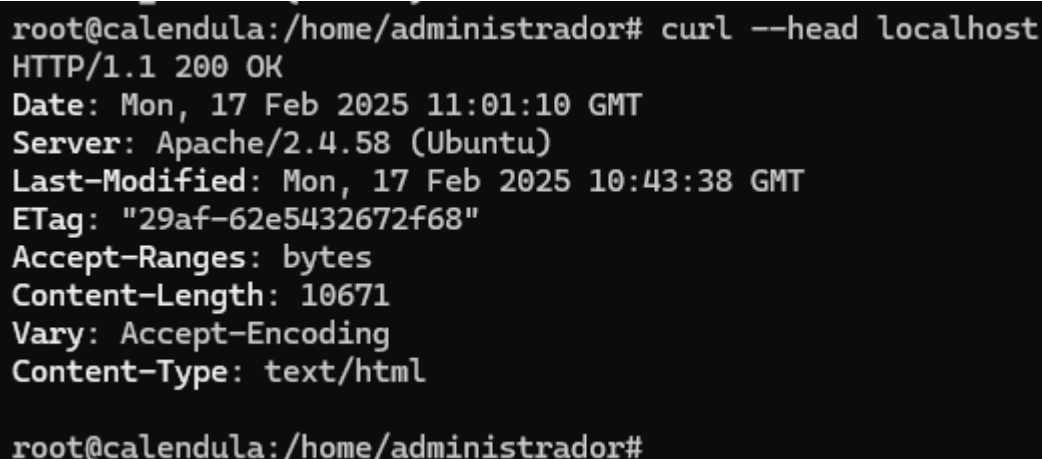
## 1. Server Hardening

---

En esta práctica, nos hemos enfocado en mejorar la seguridad de un servidor Apache aplicando distintas configuraciones para minimizar riesgos y fortalecer su protección frente a ataques comunes.

### 1.1 Configuración de Seguridad en Apache

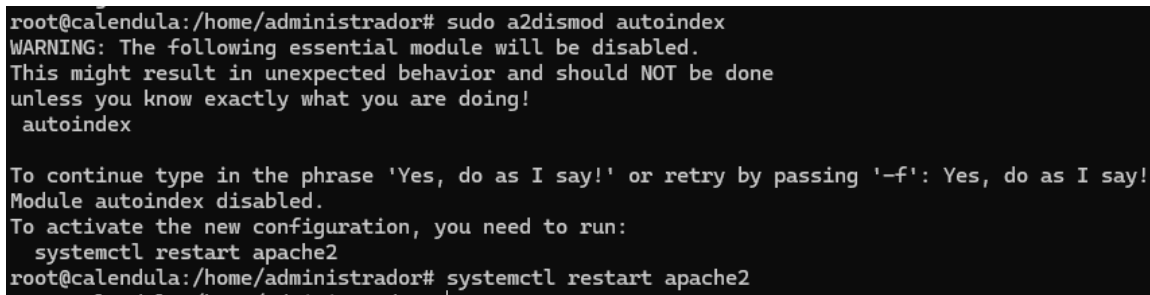
El primer paso fue reducir la superficie de ataque eliminando los módulos de Apache que no son esenciales para el funcionamiento del servidor. Entre los módulos deshabilitados, destaca `mod autoindex`, que genera listados automáticos de directorios cuando no hay un archivo de índice presente, lo que podría exponer información sensible.



```
root@calendula:/home/administrador# curl --head localhost
HTTP/1.1 200 OK
Date: Mon, 17 Feb 2025 11:01:10 GMT
Server: Apache/2.4.58 (Ubuntu)
Last-Modified: Mon, 17 Feb 2025 10:43:38 GMT
ETag: "29af-62e5432672f68"
Accept-Ranges: bytes
Content-Length: 10671
Vary: Accept-Encoding
Content-Type: text/html

root@calendula:/home/administrador#
```

Figura 1: Curl head.



```
root@calendula:/home/administrador# sudo a2dismod autoindex
WARNING: The following essential module will be disabled.
This might result in unexpected behavior and should NOT be done
unless you know exactly what you are doing!
 autoindex

To continue type in the phrase 'Yes, do as I say!' or retry by passing '-f': Yes, do as I say!
Module autoindex disabled.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@calendula:/home/administrador# systemctl restart apache2
```

Figura 2: Deshabilitar autoindex.

Esta imagen muestra la habilitación del módulo `modheaders`, que nos permite modificar y configurar cabeceras HTTP en las respuestas del servidor. Esto es fundamental para implementar medidas de seguridad como HSTS (Strict

Transport Security) y CSP (Content Security Policy), las cuales mejoran la protección frente a ataques de interceptación y ejecución de scripts maliciosos.

```
root@calendula:/home/administrador# sudo a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
```

Figura 3: Habilitar header.

```
GNU nano 7.2 /etc/apache2/sites-available/default-ssl.conf
<VirtualHost *:443>
    ServerAdmin webmaster@localhost

    Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains"

    DocumentRoot /var/www/html
```

Figura 4: Configuración default.

Es importante analizar la configuración inicial para determinar qué ajustes deben aplicarse y qué elementos pueden representar riesgos de seguridad.

```
GNU nano 7.2 /etc/apache2/conf-available/security.conf *
# Changing the following options will not really affect the security of the
# server, but might make attacks slightly more difficult in some cases.
Header set Content-Security-Policy "default-src 'self'; img-src *; media-src medial.com media2.com; script-src userscripts.example.com"
#
```

Figura 5: Header set en security.conf..

## 1.2 Implementación de ModSecurity

Este archivo contiene una configuración base que nos ayuda a establecer reglas de seguridad en el servidor. A partir de esta configuración, se pueden personalizar reglas adicionales para reforzar la protección contra ataques web.

```
administrador# sudo cp /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf
```

Figura 7: Copia de .conf recomendada

```
root@calendula:/home/administrador# curl localhost/index.html?testparam=test
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access this resource.</p>
</body></html>
root@calendula:/home/administrador#
```

Figura 8: Prueba curl denegada.

Esta imagen demuestra cómo, tras la activación de ModSecurity, las solicitudes que antes eran aceptadas ahora son bloqueadas por el firewall. En este caso, intentamos hacer una petición HTTP que no cumple con las reglas de seguridad, obteniendo como resultado un error 403 Forbidden, lo que indica que la solicitud ha sido correctamente filtrada por el WAF.

### 1.3 Implementación de Reglas OWASP en ModSecurity

```
root@calendula:/home/administrador# git clone https://github.com/SpiderLabs/owasp-modsecurity-crs.git
Cloning into 'owasp-modsecurity-crs'...
```

*Figura 9: Git clone owasp.*

Primero clonamos del repositorio oficial de OWASP ModSecurity Core Rule Set (CRS). Estas reglas predefinidas ayudan a detectar y mitigar ataques como SQL Injection, Cross-Site Scripting (XSS) y ejecución remota de comandos (RCE).

```
root@calendula:/home/administrador/owasp-modsecurity-crs# sudo mv crs-setup.conf.example /etc/modsecurity/crs-setup.conf
root@calendula:/home/administrador/owasp-modsecurity-crs#
```

*Figura 10: Mv crs.setup.example.*

Renombramos el archivo crs-setup.conf.example a crs-setup.conf y lo movemos a la carpeta de configuración de ModSecurity. Este archivo contiene las configuraciones iniciales del conjunto de reglas OWASP, permitiendo su activación en nuestro servidor.

```
root@calendula:/home/administrador/owasp-modsecurity-crs# sudo mv rules/ /etc/modsecurity
```

*Figura 11: Renombrar las reglas*

Realizamos la configuración final de las reglas de ModSecurity asegurándonos de que sean reconocidas y aplicadas correctamente por Apache. Renombramos y organizamos los archivos de reglas dentro de la estructura de configuración del servidor.

```

GNU nano 7.2 /etc/apache2/mods-enabled/security2.conf *
<IfModule security2_module>
    # Default Debian dir for modsecurity's persistent data
    SecDataDir /var/cache/modsecurity
    SecRuleEngine On
    # Include all the *.conf files in /etc/modsecurity.
    # Keeping your local configuration in that directory
    # will allow for an easy upgrade of THIS file and
    # make your life easier
    IncludeOptional /etc/modsecurity/*.conf

    # Include OWASP ModSecurity CRS rules if installed
    IncludeOptional /usr/share/modsecurity-crs/*.load
</IfModule>

```

Figura 12: security2.

Esto garantiza que Apache procese todas las solicitudes siguiendo las políticas de seguridad establecidas.

```

GNU nano 7.2 /etc/apache2/sites-available/000-default.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/html
    SecRuleEngine On
    SecRule ARGS:testparam "@contains test" "id:1234,deny,status:403,msg:'Cazado por Ciberseguridad'"
    # 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

    # For most configuration files from conf-available/, which are
    # enabled or disabled at a global level, it is possible to
    # include a line for only one particular virtual host. For example the
    # following line enables the CGI configuration for this host only
    # after it has been globally disabled with "a2disconf".
    #Include conf-available/serve-cgi-bin.conf
</VirtualHost>

```

Figura 13: apache default modificado.

Aquí presentamos la versión modificada de la configuración por defecto de Apache, integrando todas las mejoras de seguridad implementadas hasta este

punto. Se incluyen reglas adicionales y ajustes que refuerzan la protección contra ataques web.

```
root@calendula:/home/administrador# curl localhost/index.html?testparam=test
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access this resource.</p>
</body></html>
root@calendula:/home/administrador#
```

Figura 14: Prueba curl denegada.

Volvemos a validar el funcionamiento de las reglas OWASP en ModSecurity. Al ejecutar una solicitud con un parámetro sospechoso, el firewall responde con un error 403 Forbidden, indicando que la petición ha sido identificada como potencialmente maliciosa y ha sido bloqueada.

## 1.4 Protección Contra Ataques DoS con mod\_evasive

```
root@calendula:/etc/modsecurity/owasp-modsecurity-crs# sudo apt install libapache2-mod-evasive
```

Figura 15: Descarga del mod evasive.

Instalamos el módulo mod\_evasive, que ayuda a mitigar ataques de denegación de servicio (DoS) detectando múltiples solicitudes sospechosas provenientes de la misma IP.

```
GNU nano 7.2 dockerfile
# Usar una imagen base de Ubuntu
FROM ubuntu:latest

# Instalar Apache y mod_evasive
RUN apt update && apt install -y apache2 libapache2-mod-evasive

# Habilitar el módulo mod_evasive
RUN a2enmod evasive

# Configurar Apache para escuchar en los puertos 80 y 443
EXPOSE 80 443

# Reiniciar Apache
CMD ["apache2ctl", "-D", "FOREGROUND"]
```

Figura 16: Dockerfile para la instalación de apache con el mod\_evasive.



```

GNU nano 7.2 /etc/apache2/mods-available/evasive.conf *
<IfModule mod_evasive20.c>
    DOSHashTableSize    3097
    DOSPageCount        20
    DOSSiteCount         50
    DOSPageInterval     1
    DOSSiteInterval     1
    DOSBlockingPeriod   10

    #DOSEmailNotify      you@yourdomain.com
    #DOSSystemCommand    "su - someuser -c '/sbin/... %s ...'"
    DOSLogDir            "/var/log/mod_evasive"
</IfModule>

```

Figura 17: Configuración del mod\_evasive.

Esta imagen muestra los parámetros de configuración de mod\_evasive, donde establecemos el umbral de solicitudes permitidas antes de que una IP sea bloqueada. Ajustamos valores como el número de accesos por segundo y el tiempo de bloqueo de direcciones IP sospechosas.

```

alendula:~# ab -n 1000 -c 100 http://localhost/

```

Figura 18: Testeo con apache bench.

Para verificar la efectividad de mod\_evasive, realizamos una prueba de carga utilizando Apache Bench. Este test simula múltiples peticiones concurrentes al servidor para evaluar si el módulo responde bloqueando solicitudes excesivas.

```

Time taken for tests:    0.988 seconds
Complete requests:      1000
Failed requests:         937
  (Connect: 0, Receive: 0, Length: 937, Exceptions: 0)
Non-2xx responses:      937
Total transferred:      1114933 bytes
HTML transferred:       929011 bytes
Requests per second:    1011.84 [#/sec] (mean)
Time per request:       98.830 [ms] (mean)
Time per request:       0.988 [ms] (mean, across all concurrent requests)
Transfer rate:          1101.69 [Kbytes/sec] received

Connection Times (ms)
  min   mean[+/-sd] median   max
Connect:    0     1   2.2      0     9
Processing:  4    96  23.1     92   194
Waiting:    0    50  22.0     48   158
Total:      10    97  23.0     92   199

Percentage of the requests served within a certain time (ms)
 50%    92
 66%    95
 75%   100
 80%   105
 90%   130
 95%   148
 98%   156
 99%   165
100%   199 (longest request)
sent@alendula:~#

```

Figura 19: Resultado apache bench.

Se confirma que `mod_evasive` detecta el tráfico anómalo y responde bloqueando las IPs que exceden el umbral de solicitudes, protegiendo así el servidor contra ataques de denegación de servicio.

## 1.5 Instalar un certificado digital en el servidor Apache.

En esta práctica, hemos instalado un certificado SSL auto-firmado en nuestro servidor Apache, lo que permite cifrar el tráfico entre los clientes y el servidor. Aunque un certificado auto-firmado no es validado por una autoridad de certificación (CA) reconocida, proporciona una capa de seguridad útil para entornos internos o pruebas.

```
root@administrador-VirtualBox:~# sudo mkdir /etc/apache2/ssl
```

*Figura 20: Creamos el directorio ssl.*

Primero creamos el directorio `/etc/apache2/ssl`, donde almacenaremos los archivos del certificado y la clave privada. Mantener estos archivos en una ubicación dedicada dentro del sistema facilita la administración y mejora la seguridad.

```
root@administrador-VirtualBox:~# sudo openssl req -x509 -nodes -days 365 \
> -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt
.+.....+.+...+++++++
*.+.....+.+..+.....+.+..+++++++
+++++++*....+.+.....+
```

*Figura 21: Generamos la clave y certificado.*

Aquí evidenciamos la ejecución del comando openssl para generar un certificado auto-firmado y su clave privada. Se especifican los siguientes parámetros:

- -x509: Crea un certificado en lugar de una solicitud de firma.
- -nodes: Omite la protección con contraseña para facilitar el arranque automático de Apache.
- -days 365: Establece una validez de un año.
- -newkey rsa:2048: Genera una clave privada RSA de 2048 bits junto con el certificado.

Tras la ejecución del comando, se solicitan datos como el nombre del dominio (Common Name), país, ciudad y correo electrónico. Una vez completado este paso, los archivos generados (apache.crt y apache.key) quedan almacenados en /etc/apache2/ssl/.

```

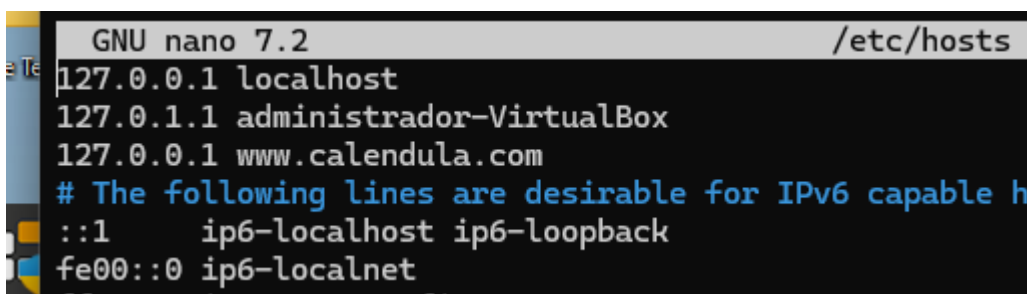
GNU nano 7.2 /etc/apache2/sites-available/default-ssl.conf *
<IfModule mod_ssl.c>
  <VirtualHost _default_:443>
    ServerAdmin admin@calendula.com
    ServerName www.calendula.com
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/apache.crt
    SSLCertificateKeyFile /etc/apache2/ssl/apache.key
    <FilesMatch "\.(cgi|shtml|phtml|php)$">
      SSLOptions +StdEnvVars
    </FilesMatch>
    <Directory /usr/lib/cgi-bin>
      SSLOptions +StdEnvVars
    </Directory>
    BrowserMatch "MSIE [2-6]" \
      nokeepalive ssl-unclean-shutdown \
      downgrade-1.0 force-response-1.0
    BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown
  </VirtualHost>
</IfModule>

```

Figura 22: Configuración de apache.

Modificamos el archivo /etc/apache2/sites-available/default-ssl.conf para incluir las rutas del certificado y la clave privada.

Además, activamos el módulo SSL con `a2enmod ssl` y habilitamos el sitio con `a2ensite default-ssl.conf`. Finalmente, reiniciamos Apache para aplicar los cambios.



```
GNU nano 7.2 /etc/hosts
127.0.0.1 localhost
127.0.1.1 administrador-VirtualBox
127.0.0.1 www.calendula.com
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
```

Figura 23: Etc/hosts modificado.

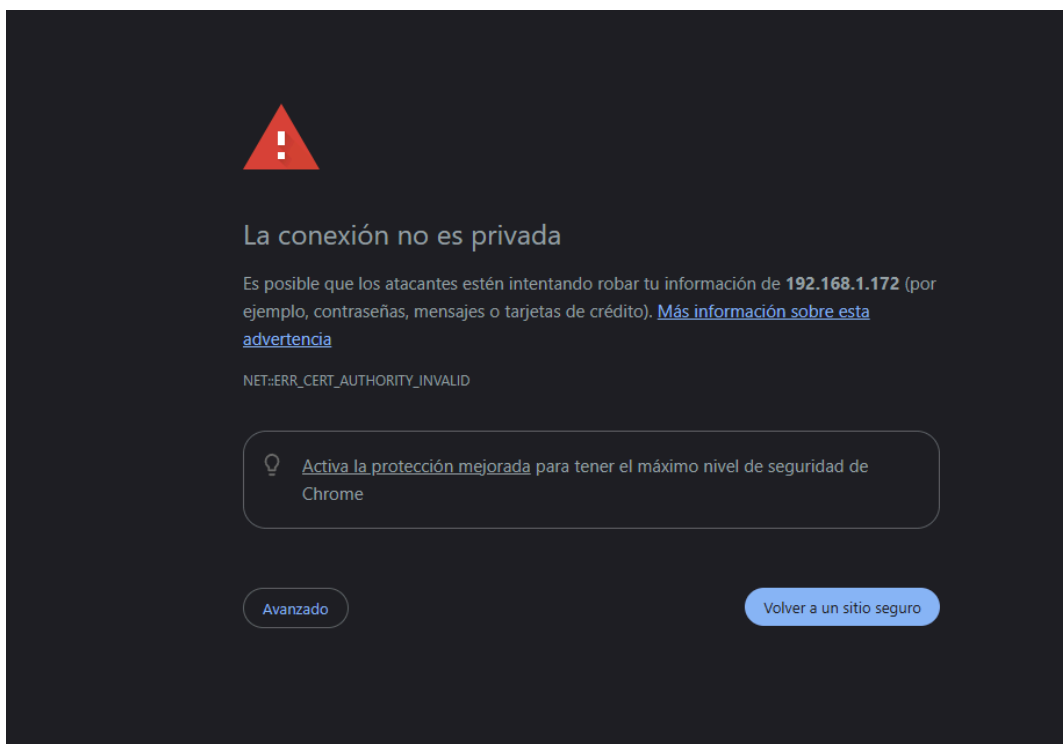
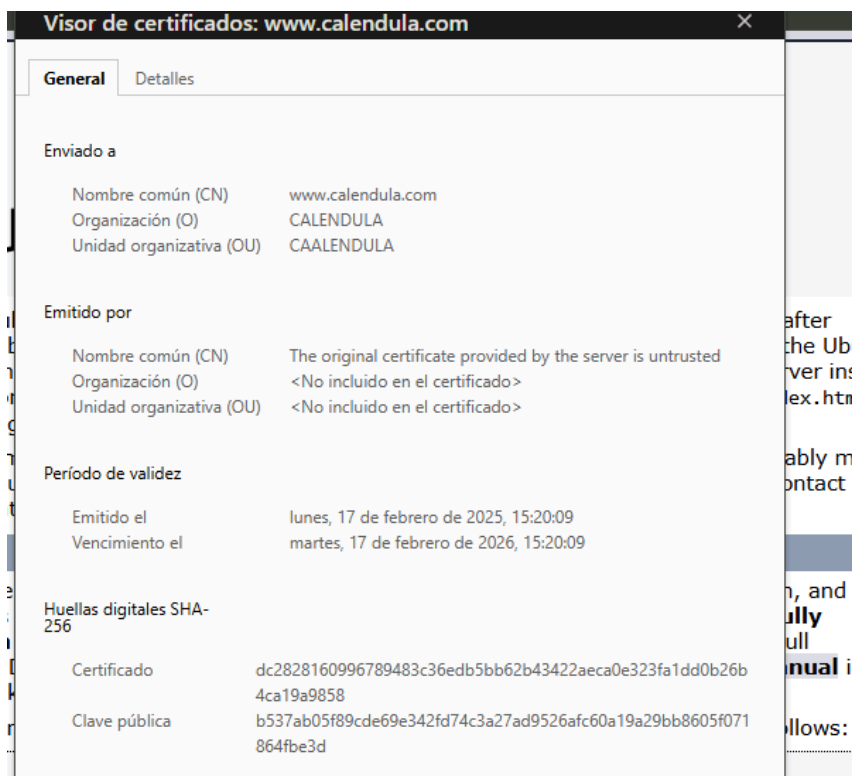


Figura 24: Acceso no privado.

Accedemos al sitio mediante `https://` y el navegador muestra una advertencia indicando que la conexión no es privada. Esto ocurre porque el certificado no está firmado por una autoridad de confianza. Sin embargo, podemos continuar de forma manual añadiendo una excepción en el navegador.



*Figura 25: Comprobación del certificado.*

Aquí verificamos que el certificado SSL está funcionando correctamente. En la interfaz del navegador, inspeccionamos el certificado y confirmamos que los detalles coinciden con los valores ingresados durante la generación del certificado.



*Figura 26: Redirección a la página segura.*

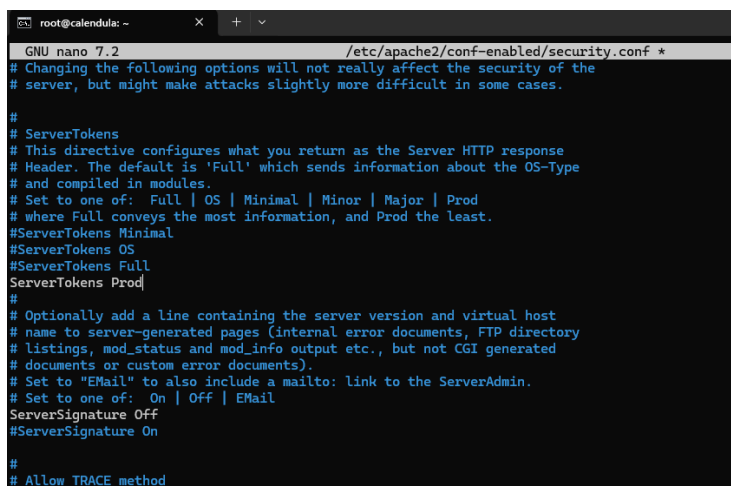
En este paso, configuramos una redirección automática desde HTTP (puerto 80) a HTTPS (puerto 443). Para ello, editamos el archivo de configuración de Apache y añadimos la siguiente directiva en el bloque del puerto 80.

Esto garantiza que cualquier usuario que intente acceder al sitio sin cifrado sea automáticamente redirigido a la versión segura.

## 1.6 Apache hardening best practices.

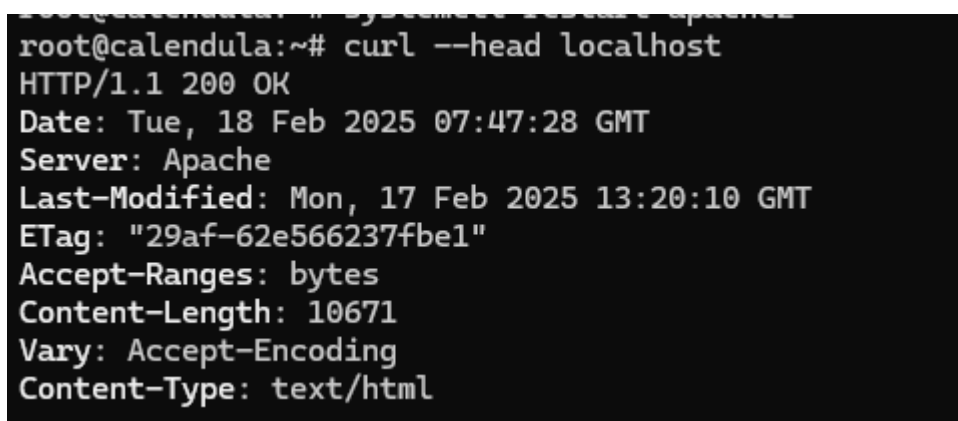
### 1.6.1 Remove Server Version Banner

Uno de los primeros pasos para fortalecer la seguridad de Apache es eliminar la información del banner del servidor. De manera predeterminada, Apache muestra detalles sobre su versión y el sistema operativo en los encabezados de respuesta HTTP, lo que puede facilitar a los atacantes la identificación de vulnerabilidades específicas. Para evitar esto, modificamos el archivo `httpd.conf`, agregando las directivas `ServerTokens Prod` y `ServerSignature Off`. Luego, reiniciamos Apache para aplicar los cambios. Con esto, evitamos exponer información sensible sobre nuestra infraestructura.



```
GNU nano 7.2 /etc/apache2/conf-enabled/security.conf *
# Changing the following options will not really affect the security of the
# server, but might make attacks slightly more difficult in some cases.
#
# ServerTokens
# This directive configures what you return as the Server HTTP response
# Header. The default is 'Full' which sends information about the OS-Type
# and compiled in modules.
# Set to one of: Full | OS | Minimal | Minor | Major | Prod
# where Full conveys the most information, and Prod the least.
#ServerTokens Minimal
#ServerTokens OS
#ServerTokens Full
ServerTokens Prod
#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (internal error documents, FTP directory
# listings, mod_status and mod_info output etc., but not CGI generated
# documents or custom error documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of: On | Off | EMail
ServerSignature Off
#ServerSignature On
#
# Allow TRACE method
```

Figura 27: Eliminar información en el banner.



```
root@calendula:~# curl --head localhost
HTTP/1.1 200 OK
Date: Tue, 18 Feb 2025 07:47:28 GMT
Server: Apache
Last-Modified: Mon, 17 Feb 2025 13:20:10 GMT
ETag: "29af-62e566237fbel"
Accept-Ranges: bytes
Content-Length: 10671
Vary: Accept-Encoding
Content-Type: text/html
```

Figura 28: Resultado del cambio de configuración.

### 1.6.2 Disable directory browser listing

Si la navegación de directorios está habilitada, los usuarios pueden ver la estructura de archivos y carpetas dentro del servidor web, lo que representa un riesgo de seguridad. Para prevenir esto, modificamos la configuración de Apache en el archivo `httpd.conf`, estableciendo `Options -Indexes` en la directiva correspondiente al directorio raíz o a los directorios que queramos proteger. Tras reiniciar Apache, cualquier intento de acceder directamente a una carpeta sin un archivo de índice mostrará un error en lugar de la lista de archivos disponibles.

```
root@calendula:/var/www/html# mkdir test
root@calendula:/var/www/html# cd test/
root@calendula:/var/www/html/test# touch hola
root@calendula:/var/www/html/test# touch quetal
root@calendula:/var/www/html/test# |
```

Figura 29: Creación de archivos de prueba.

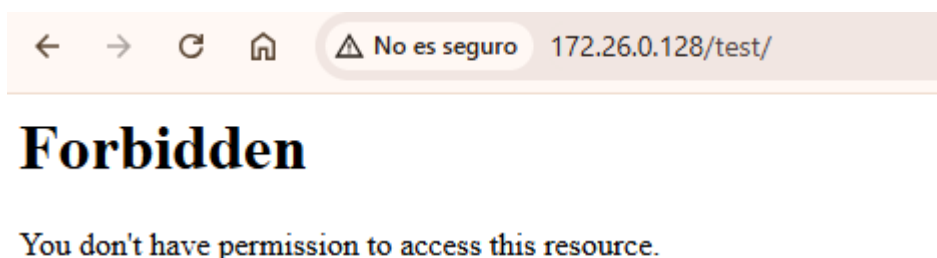


Figura 30: Confirmación del acceso denegado.

### 1.6.3 Etag

La cabecera ETag puede filtrar información sensible como números de inodo, límites de MIME multipart y procesos secundarios del servidor. Esto podría facilitar ataques dirigidos, como la identificación de recursos en caché para evadir ciertas protecciones. Para mitigar este riesgo, configuramos la directiva `FileETag None` en el archivo `httpd.conf` y reiniciamos Apache. Con esto, evitamos exponer información innecesaria que podría ser utilizada por atacantes.

```

GNU nano 7.2 /etc/apache2/conf-available/security.conf *
#
# Forbid access to version control directories
#
# If you use version control systems in your document root, you should
# probably deny access to their directories.
#
# Examples:
#
#RedirectMatch 404 /\.git
#RedirectMatch 404 /\.svn
#
# Setting this header will prevent MSIE from interpreting files as something
# else than declared by the content type in the HTTP headers.
# Requires mod_headers to be enabled.
#
#Header set X-Content-Type-Options: "nosniff"
#
# Setting this header will prevent other sites from embedding pages from this
# site as frames. This defends against clickjacking attacks.
# Requires mod_headers to be enabled.
#
#Header set Content-Security-Policy "frame-ancestors 'self';"
FileEtag None

```

Figura 31: Directiva etag.

#### 1.6.4 Run Apache from a non-privileged account

Por defecto, Apache puede ejecutarse con cuentas de usuario genéricas como nobody o daemon, lo cual no es ideal desde un punto de vista de seguridad. Para mejorar la protección, creamos un usuario y grupo específico para Apache (apache:apache) y cambiamos la propiedad de los archivos del servidor a este usuario. Luego, en httpd.conf, actualizamos las directivas User y Group para que Apache se ejecute con la cuenta sin privilegios. Esto limita el daño potencial en caso de que el servidor sea comprometido.

```

GNU nano 7.2 /etc/apache2/apache2.conf
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 5
#
# These need to be set in /etc/apache2/envvars
User apache1
Group apache
#
# HostnameLookups: Log the names of clients on just their IP addresses

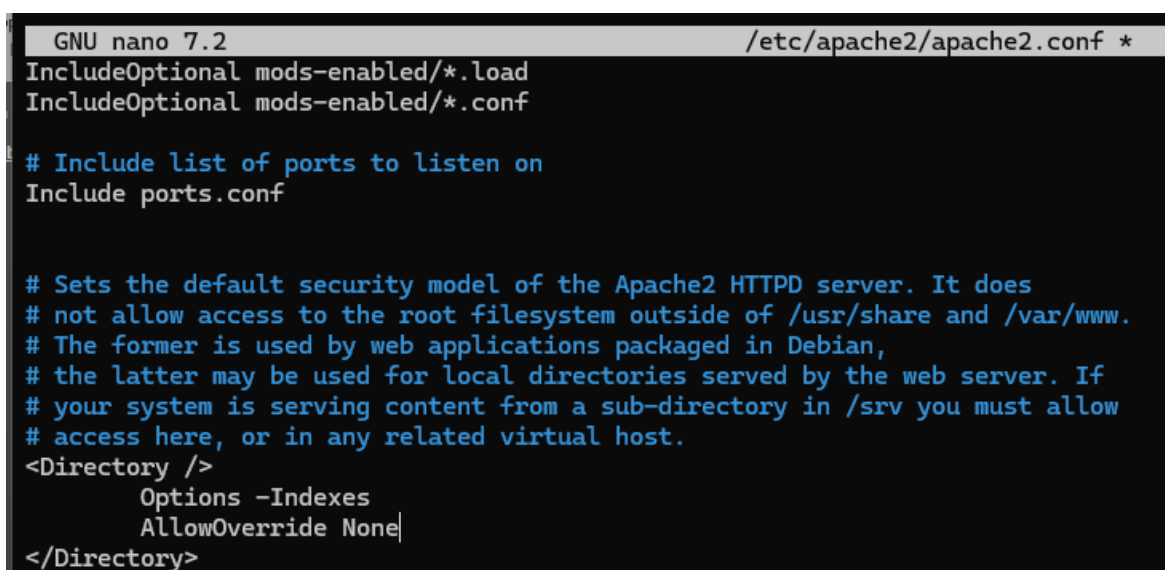
```

Figura 32: Usuario para correr apache.



### 1.6.5 System Settings Protection

Apache permite a los usuarios sobrescribir la configuración mediante archivos .htaccess, lo que puede suponer un riesgo si no se controla adecuadamente. Para evitar modificaciones no autorizadas en la configuración del servidor, establecemos AllowOverride None en la configuración de Apache dentro del archivo httpd.conf. Esto impide que archivos .htaccess alteren la configuración global del servidor, garantizando que solo los administradores puedan modificar los parámetros críticos del sistema.



```
GNU nano 7.2 /etc/apache2/apache2.conf *
IncludeOptional mods-enabled/*.load
IncludeOptional mods-enabled/*.conf

# Include list of ports to listen on
Include ports.conf

# Sets the default security model of the Apache2 HTTPD server. It does
# not allow access to the root filesystem outside of /usr/share and /var/www.
# The former is used by web applications packaged in Debian,
# the latter may be used for local directories served by the web server. If
# your system is serving content from a sub-directory in /srv you must allow
# access here, or in any related virtual host.
<Directory />
    Options -Indexes
    AllowOverride None
</Directory>
```

Figura 33: No permitir sobreescritura en apache.

## 2. Webgrafía

- <https://psegarrac.github.io/Ciberseguridad-PePS/tema1/practicas/2020/11/08/P1-SSL.html>
- <https://psegarrac.github.io/Ciberseguridad-PePS/tema3/seguridad/web/2021/03/01/Hardening-Servidor.html>
- <https://geekflare.com/cybersecurity/apache-web-server-hardening-security/>
- <https://github.com/JaumeTell/RA3/tree/main>

### **3. Conclusión**

---

La implementación de estas medidas de seguridad ha permitido reforzar la protección del servidor Apache, reduciendo riesgos asociados a ataques comunes. Aunque un certificado SSL auto-firmado proporciona cifrado, no es adecuado para entornos de producción pública, donde es recomendable adquirir un certificado de una autoridad de certificación (CA) reconocida. Además, es importante complementar estas configuraciones con auditorías regulares y actualizaciones para mantener un nivel de seguridad óptimo en el servidor.