# Servidores web

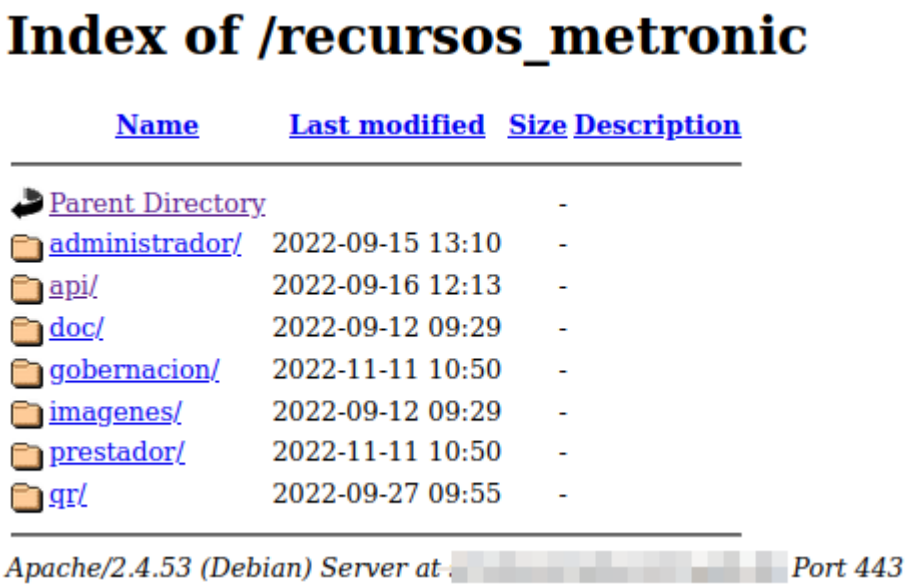
Capítulo destinado a la solución/mitigación de vulnerabilidades en servidores web.

- Listado de directorio (Index Of)
- Certificados SSL/TLS
- Eliminar la visualización de archivos de configuración
- Cross-Site Request Forgery CSRF
- Copias de seguridad
- Contenido por defecto
- Implementación de cabeceras de seguridad
- Quitar archivos de configuración

# Listado de directorio (Index Of)

Una lista de directorios se expone de forma inapropiada, lo que proporciona información potencialmente confidencial a los atacantes.

Una lista de directorio proporciona a un atacante el índice completo de todos los recursos ubicados dentro del directorio. Los riesgos y consecuencias específicos varían según los archivos enumerados y accesibles.




# Deshabilitar Index Of en Apache

Las siguientes configuraciones fueron realizadas en un servidor Debian 9, Apache 2.4 y usuario con privilegios sudo. Dependiendo del caso puede elegir una de las siguientes opciones para deshabilitar el listado de directorio.

## DEBIAN

## 1. Deshabilitar el módulo autoindex

Es el método más efectivo, pero se tiene que hacer un análisis previo para aplicarlo, ya que deshabilita esta funcionalidad a nivel global, es decir, si se quiere compartir archivos por HTTP (No recomendado) ya no se podría hacer, y los cambios afectarían a todas las aplicaciones bajo el servidor.

```
$ sudo a2dismod autoindex
```

Después de ejecutar se mostrará el mensaje de advertencia al cual responder con la siguiente frase:

```
To continue type in the phrase 'Yes, do as I say!' or retry by passing '-f': Yes, do as I say!
```

Reiniciar el servidor

Reiniciar el servidor

```
$ sudo systemctl restart apache2.service
```

## 2. Deshabilitar por archivo de configuración del sitio

Este método es el recomendable, ya que permite deshabilitar esta funcionalidad solo para el sitio en cuestión. Por ejemplo si se tiene el sitio www.sitio-de-prueba.com con el archivo de configuración (virtualhost) `sitio-de-prueba.conf`. Agregar en el archivo la siguiente directiva:

```
<VirtualHost *:80>

......

<Directory /var/www/sitio-de-prueba>
    AllowOverride All
    Options -Indexes
</Directory>

.......

</VirtualHost>
```

Guardar y recargar la configuración.

```
$ sudo systemctl reload apache2.service
```

## 3. Deshabilitar a través del archivo .htaccess

Es una alternativa similar al archivo de configuración y se tiene modificar o crear dependiendo del caso el archivo .htaccess y agregar:

```
Options -Indexes
```

Guardar el archivo y reiniciar el servidor.

```
$ sudo systemctl restart apache2.service
```

El resultado de aplicar una de las configuraciones anteriores, será la restricción de listar archivos y carpetas.

# RHE/CENTOS

## 1. Deshabilitar por archivo de configuración

Este método es el recomendable, ya que permite deshabilitar esta funcionalidad solo para el sitio en cuestión. Por ejemplo si se tiene el sitio www.sitio-de-prueba.com su correspondiente configuración está en el archivo de configuración `httpd.conf` ubicado en `/etc/httpd/conf/httpd.conf` Agregar en la configuración del virtualhost correspondiente al sitio la siguiente directiva:

```
<VirtualHost *:80>

......

<Directory /var/www/html/sitio-de-prueba>
    AllowOverride All
    Options -Indexes
</Directory>

.......

</VirtualHost>
```

Guardar y reiniciar el servidor.

```
$ sudo /etc/init.d/httpd restart
```

## 3. Deshabilitar a través del archivo .htaccess

Para esta opción se tiene que habilitar el módulo `rewrite`. Es una alternativa similar al archivo de configuración y se tiene modificar o crear dependiendo del caso el archivo .htaccess y agregar:

```
Options -Indexes
```

Guardar el archivo y probar la configuración.

# CPANEL

En ocasiones las páginas web de las entidades están publicadas en un servicio de web hosting al acceden desde el cpanel, y la configuración por defecto mantiene habilitado los índices en los directorios del cpanel y por ende en la capeta `public_html`.

1. Ingresar a cpanel, buscar y seleccionar la opción `Índices`. file 2. Seleccionar la carpeta `public_html`. file 3. Seleccionar la opción `Sin índice`. file

Aplicar la misma configuración en el resto de carpetas del sitio, ya que desde cpanel no es posible aplicar una configuración global que deshabilite el index of.

Tambien se puede aplicar las siguientes opciones:

## A. Desde el cPanel

1. Accedemos al panel de control (cPanel) y buscamos el menú de opciones Herramientas Avanzadas.
2. Entramos en la sección Index Manager donde se nos mostrará un árbol con todos los directorios de nuestra web.
3. Navegamos hasta el directorio que deseemos proteger. Para entrar en los directorios hay que hacer click en el icono que está a la izquierda del nombre de la carpeta. Cuando encontremos la carpeta que deseamos proteger hacemos click en el nombre de la misma donde podemos marcar la opción No Indexar.

## B. Desde el archivo .htaccess

1. Introduzca la siguiente línea de código `Options -Indexes`

2. También se puede evitar que se listen unos determinados archivos en concreto según su extensión. Por ejemplo, para evitar que se listen archivos de extensión .php y .html escriba: `IndexIgnore *.php *.html`
3. Este método sirve para que se listen los directorios pero haciendo que estos aparezcan en blanco. De esta forma cualquier tipo de fichero no se mostrará a la hora de acceder a su correspondiente carpeta: `IndexIngnore *`

### C. Desde un archivo index en blanco

Esta solución es sencilla, solo se debe crear un documento en blanco llamado index.html (o index.php) para evitar el listado de los archivos cuando se acceda al directorio. El problema de este método es que se debe añadir un archivo index en blanco en cada uno de los directorios en los que no se quiere mostrar los archivos.

## TOMCAT

https://support.esri.com/en/technical-article/000010708

# Certificados SSL/TLS

Let's encrypt es un servicio que ofrece certificados SSL gratuitos a través de una API. Cerbot es un cliente ACME de Let's Encrypt, que tiene varias formas de validar el dominio, busca certificados y configura automáticamente Apache y Nginx.

## Certificados Let's Encrypt

### Instalación de Cerbot en Debian

Instalación de Cerbot:

```
sudo apt-get update
sudo apt-get install software-properties-common
sudo add-apt repository universe
sudo add-apt repository ppa:cerbot/cerbot
sudo apt-get update
sudo apt-get install cerbot
```

O use aptitude otro administrador de paquetes para su distribución.

### Versiones de Ubuntu por debajo de 16.10 yakkety

```
sudo add-apt-repository ppa:certbot/certbot
sudo apt-get update
sudo apt-get install --upgrade letsencrypt
```

### Otra distribución

Si tiene alguna otra distribución, hay instrucciones de instalación adicionales en el sitio web oficial de Cerbot. Si prescinde de un administrador de paquetes, generalmente la instalación se reduce a:

```
wget -O /usr/local/bin/certbot-auto https://dl.eff.org/certbot-auto
chmod +x /usr/local/bin/certbot-auto
ln -s /usr/local/bin/certbot-auto /usr/local/bin/certbot
```

# Crear un certificado SSL

Let's Encrypt realiza automáticamente la validción de dominio (DV). La autoridad de certificación (CA) verifica la autenticidad del dominio del servidor. Una vez que haya sido validado, la CA le emitirá certificados SSL.

Ejecutar Let's Encrypt con el siguiente comando:

```
sudo -H ./letsencrypt-auto certonly --standalone -d tudominio.com -d www.tudominio.com
```

# Eliminar la visualización de archivos de configuración

Servidores Apache

Servidores Nginx

# Cross-Site Request Forgery CSRF

La vulnerabilidad CSRF (Cross-Site Request Forgery) es un tipo de ataque que se produce cuando un usuario autenticado involuntariamente envía una solicitud malintencionada a un sitio web. Este ataque aprovecha la confianza del sitio web en la sesión activa del usuario para enviar una solicitud no autorizada, que puede provocar cambios no deseados en el estado de la cuenta del usuario, como transferencias de dinero, cambios de contraseña, eliminación de datos, entre otros.

## Solución

Implementar tokens de seguridad CSRF  en formularios HTML que se verifican en el servidor para garantizar que la solicitud sea legítima y no se haya falsificado.

Para conocer más a cerca de la vulnerabilidad y formas de mitigación consultar Cross-Site Request Forgery Prevention Cheat Sheet CSRF

# Copias de seguridad

Una copia de seguridad, también conocida como respaldo o backup, es una copia de los datos y archivos importantes de un sistema, como documentos, bases de datos, configuraciones y otros elementos críticos. El propósito de una copia de seguridad es permitir la recuperación de los datos en caso de pérdida, daño o corrupción de los originales.

El almacenamiento de las copias de seguridad es una consideración crucial para garantizar su efectividad. Algunas opciones comunes para almacenar copias de seguridad son:

1. Dispositivos de almacenamiento local: Puedes guardar las copias de seguridad en dispositivos de almacenamiento físico conectados directamente a tu sistema, como discos duros externos, NAS (Network Attached Storage) u otros medios extraibles.
2. Servidores de respaldo en la red local: Si se cuenta con una red local, se puede configurar un servidor dedicado donde almacenar las copias de seguridad de varios sistemas.
3. Servidores FTP o SFTP remotos: Puedes alamacenar copias de seguridad en servidores remotos utilizando protocolos de tranasferencia de archivos como FTP (File Transfer Protocol) o SFTP (Secure File Transfer Protocol).

## Copia de seguridad en Nginx

Para realizar copias de seguridad en Ngnix se pueden utilizar los siguientes comandos:

1. Utilizar el comando cp para copiar archivos y directorios, por ejemplo:

```
cp -r /ruta/directorio_origen /ruta/directorio_destino

En Nginx los archivos de configuración se encuentran en la ruta:

/etc/nginx/

Los archivos del DocumentRoot en la ruta:

/var/www/ o /usr/share/nginx/html/
```

2. Utilizar el comando tar para crear archivos de respaldo comprimidos, por ejemplo:

```
tar -czvf archivo.tar.gz /ruta/directorio
```

## Copia de seguridad Apache

Para realizar copias de seguridad en apache se pueden utilizar los siguientes comandos:

1. Utilizar el comando cp para copiar archivos y directorios, por ejemplo:

```
cp -r /ruta/directorio_origen /ruta/directorio_destino

En Apache los archivos de configuración se encuentran en la ruta:

/etc/apache2/

Los archivos del DocumentRoot en la ruta:

/var/www/html/
```

2. Utilizar el comando tar para crear archivos de respaldo comprimidos, por ejemplo:

```
tar -czvf archivo.tar.gz /ruta/directorio
```

Es importante mencionar que las rutas especificas pueden variar según la configuración del sistema.

# Contenido por defecto

El contenido por defecto en servidores se refiere al conjunto de archivos y directorios que se proporcionan automáticamente al instalar un servidor web en un sistema operativo. Estos archivos y directorios son parte de la configuración predeterminada y su propósito es ofrecer una página o sitio web básico cuando se accede al servidor sin haber configurado o desplegado ningún sitio específico.

En muchos casos, es deseable eliminar o modificar el contenido por defecto para evitar la exposición de información innecesaria o para personalizar la experiencia del usuario al acceder al servidor. Esto se puede lograr eliminando o reemplazando los archivos y directorios por defecto con el contenido adecuado para el sitio web.

Antes de remover los directorios y archivos de instalación por defecto se recomienda realizar una copia de seguridad de los mismos.

## Remover directorios y archivos de instalación (manuales) en Apache2

Apache generalmente incluye un directorio de documentos por defecto, `/var/www/html/` en sistemas basados en Debian/Ubuntu.

Puedes eliminar los archivos y directorios por defecto en esa ubicación par evitar que sean accesibles. Por ejemplo, en linux puedes usar el siguiente comando para eliminar el contenido por defecto:

```
sudo rm -r /var/www/html/*
```

Finalmente reiniciar el servidor Apache.

## Remover directorios y archivos de instalación (manuales) en Tomcat

Los archivos de instalación por defecto en tomcat se encuentran en `/opt/tomcat/` o en alguna otra ubicación del sistema.

Dentro del directorio de Tomcat, busca el directorio `/webapps/`. Este directorio contiene las aplicaciones web desplegadas en Tomcat.

Eliminar el directorio `/docs/`, este directorio contiene la documentación y los manuales de Tomcat, puede utilizar el siguiente comando en linux para eliminar el directorio:

```
sudo rm -r /opt/tomcat/webapps/docs
```

Finalmente reiniciar el servidor Tomcat.

# Implementación de cabeceras de seguridad

Las cabeceras de seguridad son una forma de agregar capas adicionales de seguridad a los sitios web y aplicaciones web. Son metadatos enviados por el servidor web junto con las respuestas a las solicitudes del cliente, y que proporcionan información adicional sobre cómo se debe manejar la respuesta y cómo se debe proteger la sesión del usuario.

Las cabeceras de seguridad se utilizan para implementar políticas de seguridad adicionales en la comunicación entre el cliente y el servidor web, y se pueden utilizar para mitigar una variedad de vulnerabilidades de seguridad, como ataques de inyección de código, cross-site scripting (XSS), clickjacking, sniffing de paquetes y otros.

Entre las cabeceras de seguridad más comunes se incluyen:

- Content-Security-Policy (CSP): Esta cabecera permite a los sitios web especificar qué tipos de contenido (como scripts, fuentes, imágenes, etc.) se pueden cargar y desde qué orígenes se pueden cargar. Esto puede prevenir los ataques XSS, ya que limita el alcance de los scripts maliciosos.
- Strict-Transport-Security (HSTS): Esta cabecera obliga a los navegadores web a utilizar HTTPS para todas las solicitudes al servidor. Esto protege contra ataques de sniffing de paquetes y garantiza que la comunicación entre el cliente y el servidor se realice de forma segura.
- X-Frame-Options: Esta cabecera permite a los sitios web controlar si su contenido se puede cargar en un iframe de otro dominio, lo que ayuda a prevenir ataques de clickjacking.
- X-XSS-Protection: Esta cabecera activa el filtro anti-XSS del navegador y previene ataques XSS.
- X-Content-Type-Options: Esta cabecera ayuda a prevenir ataques de sniffing de contenido, especificando el tipo de contenido que el servidor web debe enviar.

## Implementación  de cabeceras de seguridad en servidores Apache

Implementación realizada en apache 2.4 y Debian se requieren que el usuario tenga permisos de administrador (sudo):

Habilitar las cabeceras de seguridad:

```
# a2enmod headers
```

Configurar del archivo security.conf:

```
# nano /etc/apache2/conf-enabled/security.conf
```

Modificar los siguientes valores:

```
Header set X-Content-Type-Options: "nosniff"

Header always set X-Frame-Options: "SAMEORIGIN"

Header always set Referrer-Policy: "no-referrer"

Header set X-XSS-Protection: "1; mode=block"
```

Finalmente reiniciar el servidor:

```
systemctl restart apache2
```

# Implementación  de cabeceras de seguridad en servidores Nginx

Implementación realizada en Nginx y Debian

Ir al directorio `/etc/nginx/conf.d/` e ingresar al archivo de configuración `.conf` que este utilizando.

Abrir el archivo de configuración y agregar las siguientes líneas al bloque `server`:

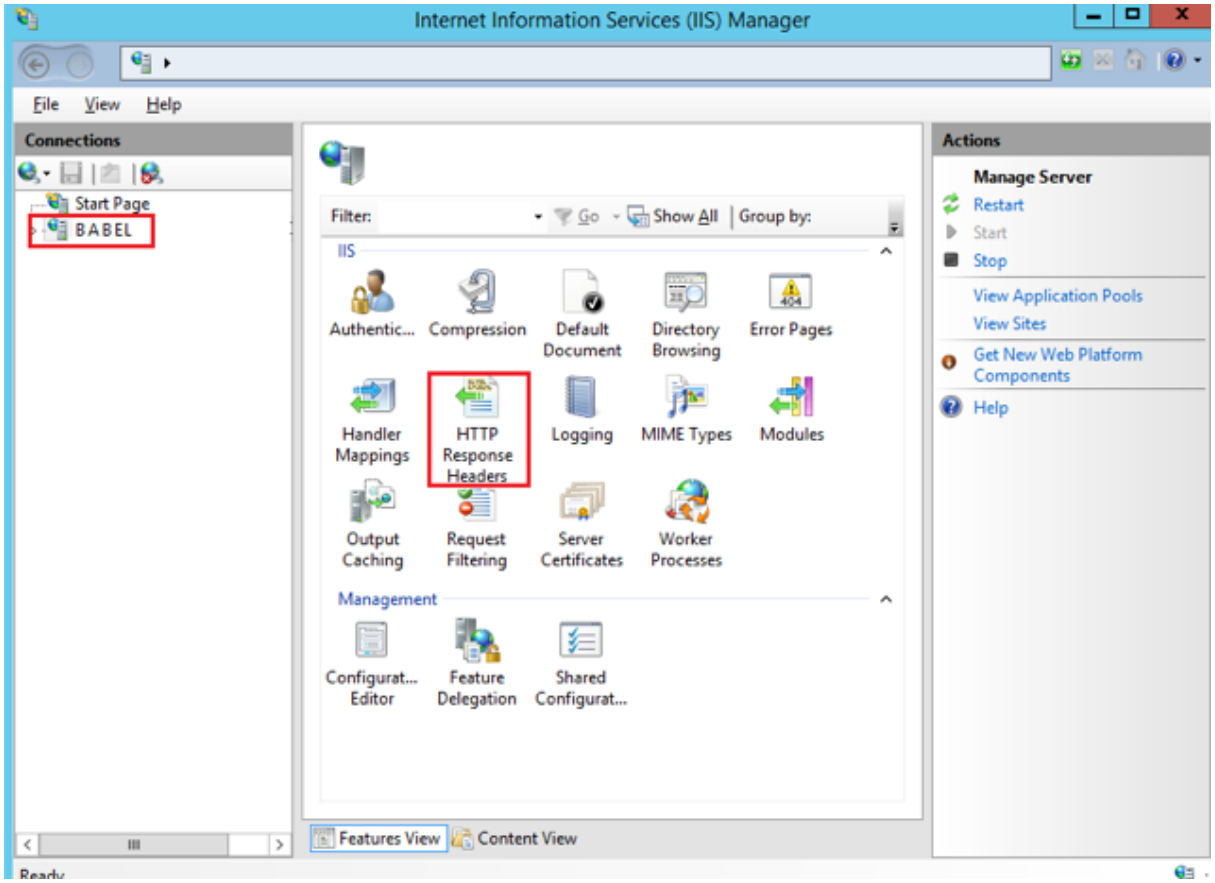
```
# Configuración de cabeceras de seguridad
add_header X-XSS-Protection "1; mode=block";
add_header X-Content-Type-Options "nosniff";
add_header X-Frame-Options "SAMEORIGIN";
add_header Referrer-Policy "strict-origin-when-cross-origin";
add_header Content-Security-Policy "default-src 'self'; script-src 'self' 'unsafe-inline'; style-src 'self' 'unsafe-inline'";
```

Reiniciar el servidor nginx:

```
sudo systemctl restart nginx
```

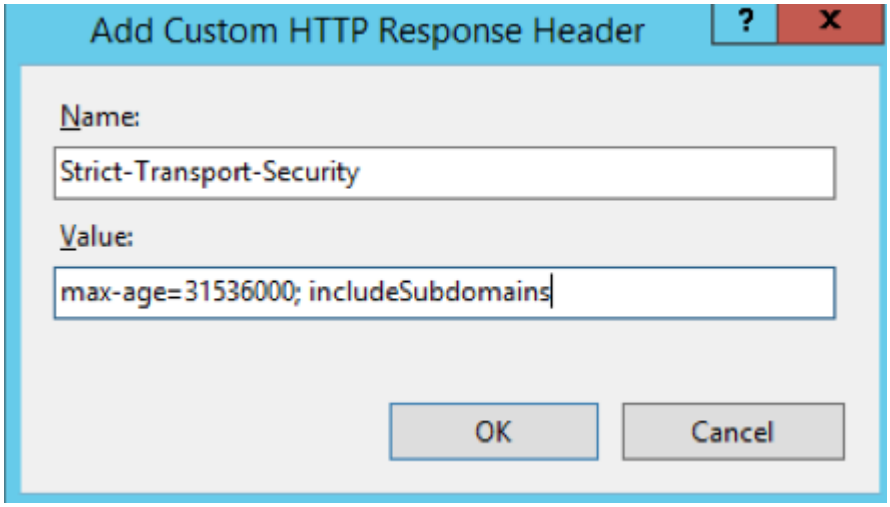
# Implementación  de cabeceras de seguridad en servidores IIS

Configuración en el Servidor IIS, para su configuración, en la ventana encabezados de respuesta HTTP, haga clic en agregar en el panel derecho de acciones y luego ingrese los detalles del encabezado como se muestra a continuación.

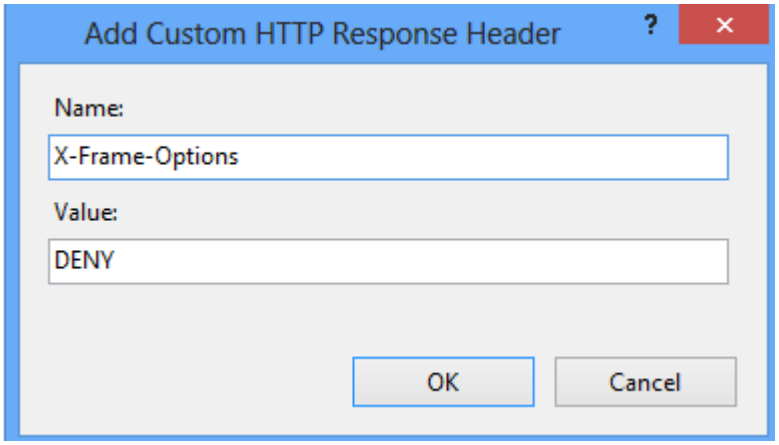



## Strict-Transport-Security

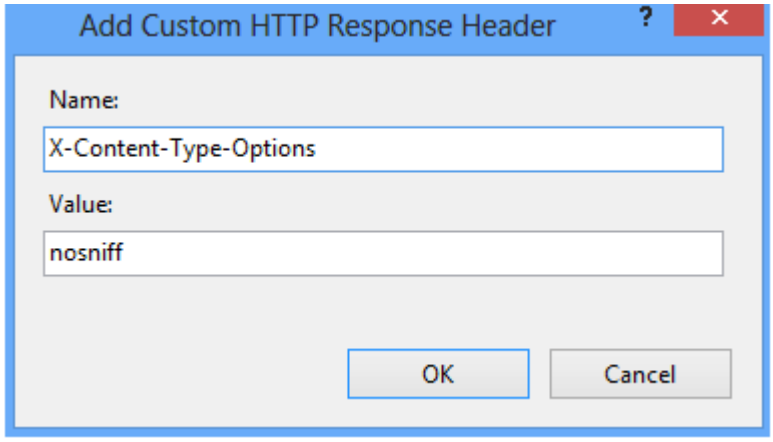
El valor “max-age=63072000” es el número de segundos que se establece para que la navegación haga uso del encabezado.



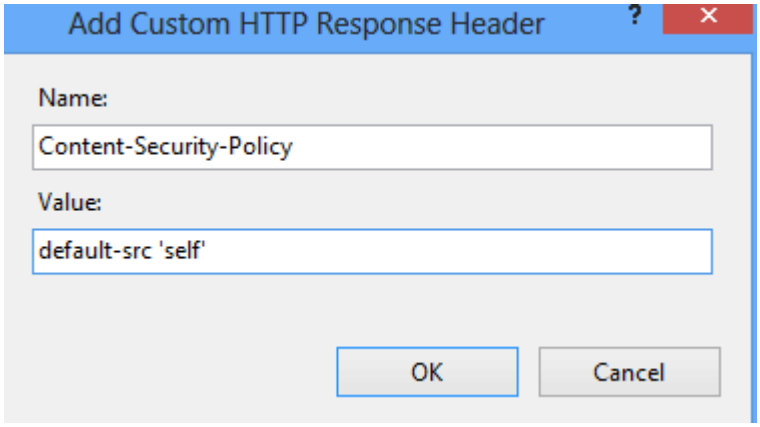

X-Frame-Options




X-Content-Type-Options




Content-Security-Policiy


Add Custom HTTP Response Header
Name:
Content-Security-Policy
Value:
default-src 'self'
OK
Cancel

# Quitar archivos de configuración

La exposición de archivos de configuración se refiere al riesgo de que los archivos de configuración de un servidor o una aplicación sean accesibles públicamente a través de Internet. Estos archivos contienen información sensible, como contraseñas, claves de API u otros datos confidenciales que podrían ser utilizados por un atacante para comprometer la seguridad del sistema.

La exposición de archivos de configuración puede ocurrir debido a una configuración incorrecta del servidor web o a la falta de medidas de seguridad adecuadas. Algunos ejemplos de archivos de configuración comunes que deben mantenerse privados son los archivos .env, .config, web.config, php.ini, entre otros.

El impacto de la exposición de archivos de configuración puede ser significativo, ya que permite a los atacantes obtener información sensible y utilizarla para realizar acciones maliciosas, como el acceso no autorizado a sistemas, la modificación de la configuración, el robo de datos o la ejecución de ataques más avanzados.

## Quitar archivos de configuración en Apache

1. Abre el archivo de configuración principal de Apache, generalmente ubicado en `/etc/httpd/httpd.conf` o `/etc/apache2/apache2.conf.`

2. Dentro del bloque Directory, puedes establecer reglas de acceso y restricciones para los archivos y directorios específicos.

```
<FilesMatch "^(.*\.pl|.*\.env|.*\.config|.*\.yml)$">
    Require all denied
</FilesMatch>
```

## Quitar archivos de configuración en Nginx

1. Abre el archivo de configuración principal de Nginx, generalmente ubicado en `/etc/nginx/nginx.conf`.

2. Dentro del bloque http, puedes agregar una directiva para denegar el acceso a los archivos específicos que deseas proteger.

```
location ~* /\.(pl|env|config|yml)$ {
    deny all;
}
```