Guía de Seguridad

Fecha de publicación: 17/02/2023

Tema: Guía de configuración de cabeceras HTTP de seguridad para servidores web

Objetivo: Proveer una guía sobre las cabeceras HTTP de seguridad, su configuración en los servidores de aplicación web más utilizados a nivel internacional y un resumen de los ataques web que se previenen mediante su uso.

Índice

Definición de cabeceras HTTP de seguridad para servidores de aplicación web

Cada vez que un usuario utiliza un navegador web para ingresar a un sitio en línea, automáticamente se establece una comunicación de red imperceptible para el usuario, donde se emite una solicitud HTTP desde el navegador hacia el servidor web, el servidor responde la solicitud del contenido solicitado, que se entrega junto con metadatos y otra información que será procesada por el navegador para manejar una serie de tareas, tales como: almacenamiento en caché de contenido, idioma y localización, codificación de caracteres, etc. Esta información se denomina cabeceras HTTP, algunas de ellas con funcionalidades de seguridad para la comunicación, ya que se relacionan con la integridad y la confidencialidad de la información enviada a través de la red. También son conocidas como encabezados de respuesta HTTP relacionados con la seguridad, los mismos modifican el comportamiento de los navegadores web para prevenir ciertas vulnerabilidades.

Su empleo se encuentra relacionado a brindar soporte de ciertas funcionalidades propias del navegador, las cabeceras HTTP de seguridad pueden ser de gran ayuda para prevenir muchos tipos de ataques comunes, incluidos *cross-site scripting* y *clickjacking*. Además, pueden proporcionar una capa adicional de seguridad para las aplicaciones web. A continuación, se describen las principales, así como los ataques a prevenir por cada una de ellas.

Principales cabeceras HTTP de seguridad para servidores de aplicación web:

X-Frame-Options

Utilizado para proteger a los usuarios contra ataques del tipo *clickjacking*, los cuales consisten en la inclusión de un elemento *iframe* transparente cargado desde un dominio externo a ser suplantado en la Web víctima; de este modo, si se colocan botones o elementos falsos encima de los botones del sitio externo original incluido, es posible engañar a la víctima para que realice acciones a través de ellos sin percatarse.

HTTP Strict Transport Security (HSTS)

El empleo de esta cabecera previene que se degrade la comunicación entre el navegador web y el servidor evitando que la comunicación pase del protocolo HTTPS (HTTP cifrado) a HTTP sin el cifrado correspondiente, obligando por el tiempo definido (en el servidor) a que siempre se navegue a través de HTTPS. La cabecera también impide que los usuarios ignoren las advertencias del navegador sobre certificados SSL/TLS no válidos o inseguros, validando la cadena de certificados desde el navegador hasta el servidor evitando ataques del tipo *Man-in-the-Middle*.

X-XSS-Protection

La aplicación de esta cabecera en el servidor de web protege contra ataques del tipo *cross-site scripting (XSS*), habilitando un filtro específico integrado del contenido servido, en la mayoría de los navegadores modernos., se recomienda habilitarlo de manera explícita y configurarlo para fortalecer la seguridad del servidor web.

Los valores válidos: 0, que deshabilita la protección, 1 que habilita la protección y 1; mode=block que le indica al navegador que bloquee la respuesta si detecta un ataque en lugar de intentar modificar el script.

Content-Security-Policy

Content-Security-Policy se trata de un conjunto de reglas que restringen, permite que se cargue contenido web en el navegador. La adopción de esta cabecera previene ataques del tipo *cross-site scripting (XSS*) e inyección de contenido que se encuentran entre los riesgos del OWASP top 10. Los ataques de XSS ocurren cuando un atacante logra comprometer un sitio web desprotegido mediante la inyección de código malicioso, cuando un usuario intenta interactuar con el sitio, el código malicioso insertado se ejecuta en el navegador del usuario víctima.

Content-Security-Policy correctamente configurado, evita que se produzcan los ataques mencionados. Limitando la carga de contenido (Javasript, xml, frames, etc) únicamente desde lugares o sitios de confianza.

X-Content-Type-Options

Esta cabecera sólo tiene un valor válido, nosniff. Ayuda a reducir el riesgo de que se produzca un ataque basado en técnicas confusión de tipos MIME y la exposición a descargas no autorizadas. En general evita que se carguen hojas de estilo o scripts maliciosos.

Cache-Control

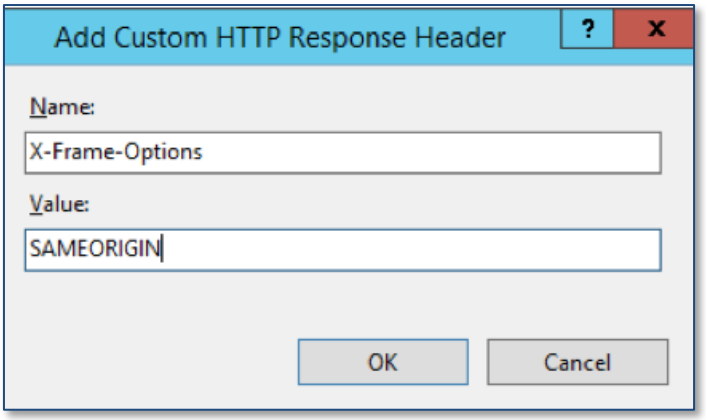
Se utiliza para especificar directivas para mecanismos de almacenamiento en caché, tanto en solicitudes como en respuestas. Las directivas de almacenamiento en caché son unidireccionales, lo que significa que una directiva dada en una solicitud no implica que se deba indicar necesariamente la misma directiva en la respuesta. Sin el uso de esta cabecera el contenido que fuese cacheado por un navegador quedará almacenado en el dispositivo del usuario.
El valor “no-cache” obliga a los cachés a enviar la solicitud al servidor de origen para su validación antes de liberar una copia en caché.

A continuación, describimos la configuración de estas cabeceras HTTP en los principales servidores web utilizados a nivel internacional:

1. Servidor IIS:

Configuración de la cabecera X-Frame-Options:

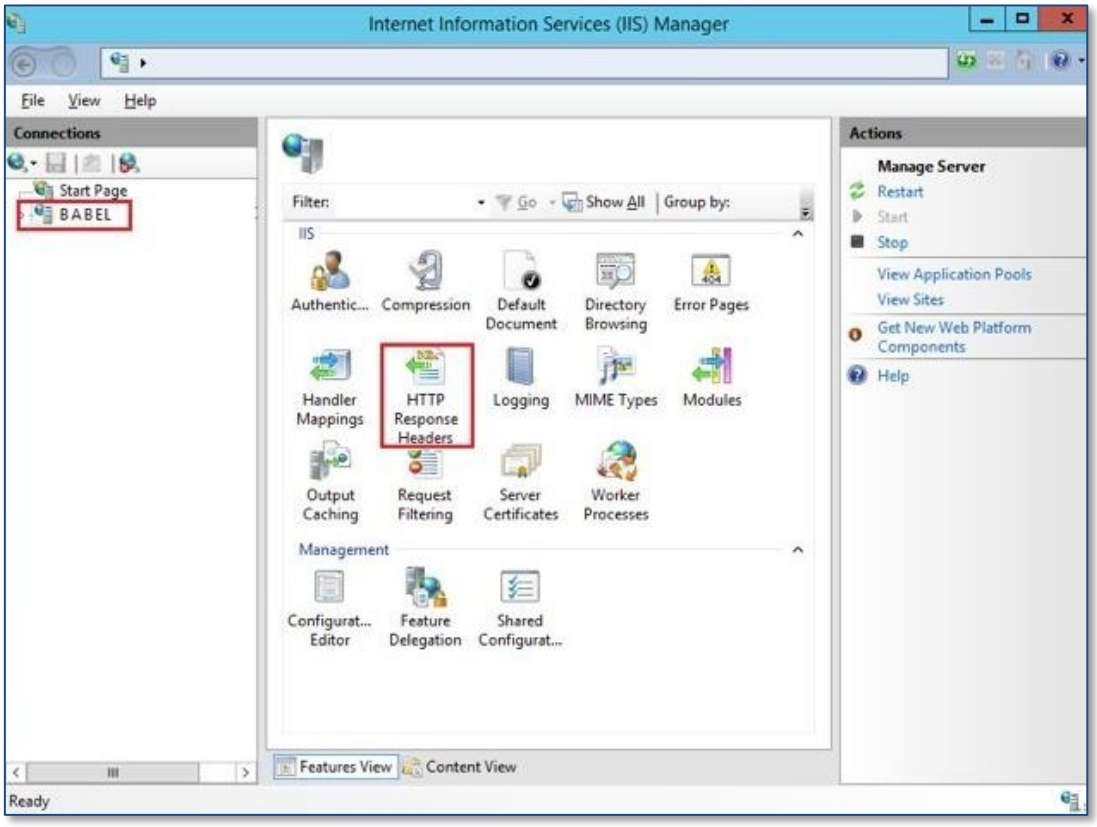
Para su configuración, en la ventana de encabezados de respuesta HTTP, hacer clic en agregar en el panel derecho de acciones y luego ingresar los detalles del encabezado como se muestra a continuación.

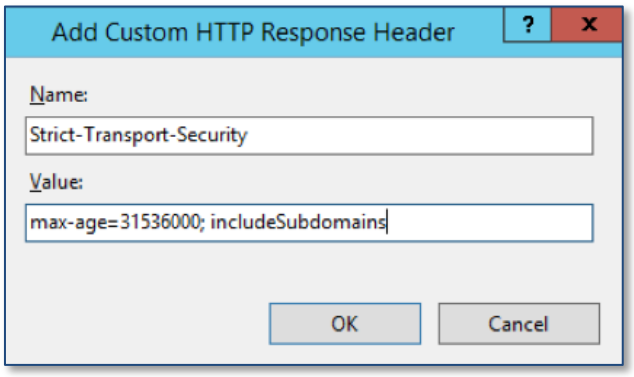


Configuración de la cabecera Strict-Transport-Security:

Para su configuración, en la ventana de encabezados de respuesta HTTP, hacer clic en agregar en el panel derecho de acciones y luego ingresar los detalles del encabezado como se muestra a continuación.

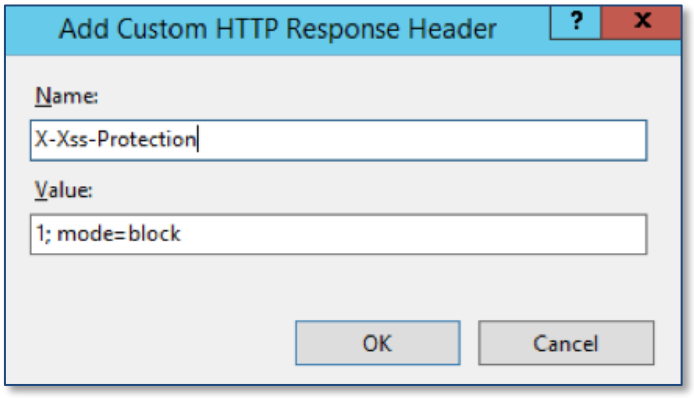
El valor “max-age=63072000” es el número de segundos que se establece para que la navegación hacer uso del encabezado.





Configuración de la cabecera X-XSS-Protection:

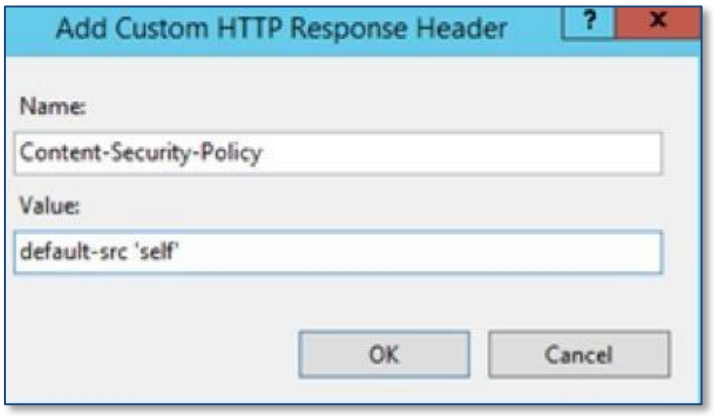
Para su configuración, en la ventana de encabezados de respuesta HTTP, hacer clic en agregar en el panel derecho de acciones y luego ingresar los detalles del encabezado como se muestra a continuación.



Configuración de la cabecera Content-Security-Policy:

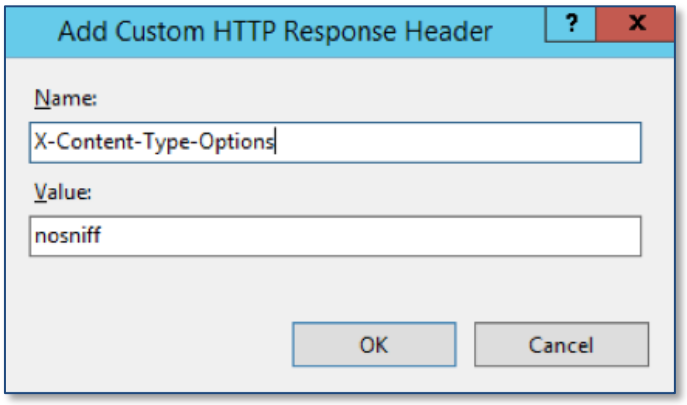
Para su configuración, en la ventana de encabezados de respuesta HTTP, hacer clic en agregar en el panel derecho de acciones y luego ingresar los detalles del encabezado como se muestra a continuación.

El valor **"default-src, self"** obliga a que todo el contenido provenga del mismo origen que el del sitio (esto excluye subdominios).



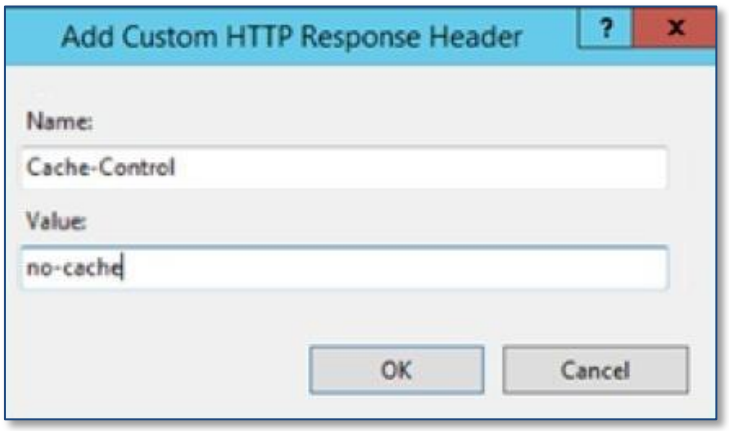
Configuración de la cabecera X-Content-Type-Options:

Para su configuración, en la ventana de encabezados de respuesta HTTP, hacer clic en agregar en el panel derecho de acciones y luego ingresar los detalles del encabezado como se muestra a continuación.



Configuración de la cabecera Cache-Control:

Para su configuración, en la ventana de encabezados de respuesta HTTP, hacer clic en agregar en el panel derecho de acciones y luego ingresar los detalles del encabezado como se muestra a continuación.



En el cuadro de diálogo establecer encabezados de respuesta HTTP comunes, marque la casilla para que caduque el contenido web inmediatamente y luego hacer clic en Aceptar.

2. Servidor Nginx:

Configuración de la cabecera X-Frame-Options:

Para realizar la configuración correspondiente, agregar lo siguiente en nginx.conf bajo el bloque del servidor y reiniciar el servidor para verificar los resultados.

```
add_header X-Frame-Options “DENY”;
```

Configuración de la cabecera Strict-Transport-Security:

Para realizar la configuración correspondiente, agregar la siguiente entrada en nginx.conf bajo el bloque SSL del servidor y reiniciar el servidor para verificar los resultados.

```
add_header Strict-Transport-Security 'max-age=31536000; includeSubDomains; preload';
```

Configuración de la cabecera X-XSS-Protection:

Para realizar la configuración correspondiente, agregar lo siguiente en nginx.conf bajo el bloque HTTP y reiniciar el servidor para verificar los resultados.

```
add_header X-XSS-Protection "1; mode = block";
```

Configuración de la cabecera Content-Security-Policy:

Se puede realizar la configuración correspondiente, a través de **default-src**, esta configuración podría indicar las fuentes admitidas para todo tipo de contenido. Por ejemplo, al utilizar "**default-src 'self';"** obliga a que todo el contenido provenga del mismo origen que del sitio (esto excluye subdominios). Esta configuración se aplicará a todos los tipos de contenido, que pueden ser diversos, desde scripts Javascript, hojas de estilo o imágenes.

Agregar lo siguiente en nginx.conf bajo el bloque del servidor y reiniciar el servidor para verificar los resultados.

```
add_header Content-Security-Policy "default-src 'self';";
```

Configuración de la cabecera X-Content-Type-Options:

Para realizar la configuración correspondiente, agregar lo siguiente en nginx.conf bajo el bloque del servidor y reiniciar el servidor para verificar los resultados.

```
add_header X-Content-Type-Options nosniff;
```

Configuración de la cabecera Cache-Control:

Para realizar la configuración correspondiente, agregar lo siguiente en nginx.conf bajo el bloque del servidor y reiniciar el servidor para verificar los resultados.

```
add_header Cache-Control "private, no-cache, no-store, max-age=0";
```

3. Servidor Apache:

Configuración de la cabecera X-Frame-Options:

Para realizar la configuración correspondiente, agregar la siguiente línea en httpd.conf y reiniciar el servidor web para verificar los resultados.

```
Header always append X-Frame-Options DENY
```

Configuración de la cabecera Strict-Transport-Security:

Para realizar la configuración correspondiente, agregar la siguiente entrada en httpd.conf y reiniciar el servidor web para verificar los resultados.

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

Configuración de la cabecera X-XSS-Protection:

Para realizar la configuración correspondiente, agregar la siguiente entrada en httpd.conf y reiniciar el servidor web para verificar los resultados.

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

Configuración de la cabecera Content-Security-Policy:

Para realizar la configuración correspondiente, agregar la siguiente línea en httpd.conf y reiniciar el servidor web para verificar los resultados.

```
Header set Content-Security-Policy "default-src 'self';"
```

Configuración de la cabecera X-Content-Type-Options:

Para realizar la configuración correspondiente, agregar la siguiente entrada en httpd.conf y reiniciar el servidor web para verificar los resultados.

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

Configuración de la cabecera Cache-Control:

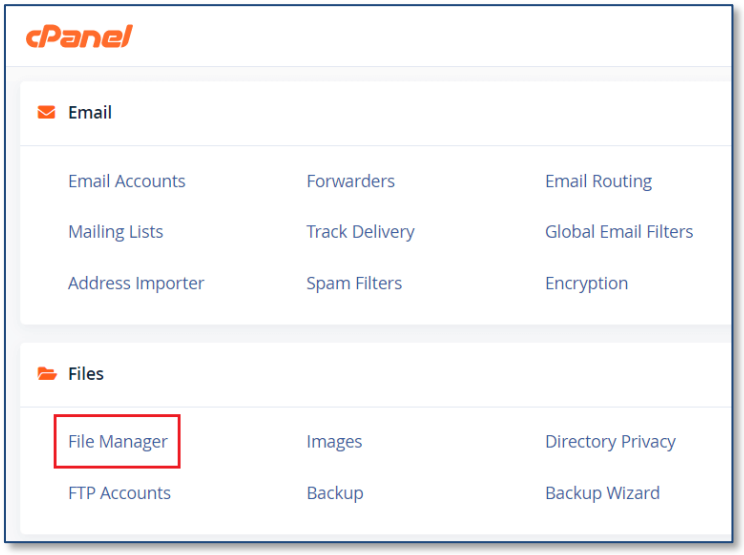
Para realizar la configuración correspondiente, agregar la siguiente entrada en httpd.conf y reiniciar el servidor web para verificar los resultados.

```
Header set Cache-Control "private, no-cache, no-store, max-age=0”
```

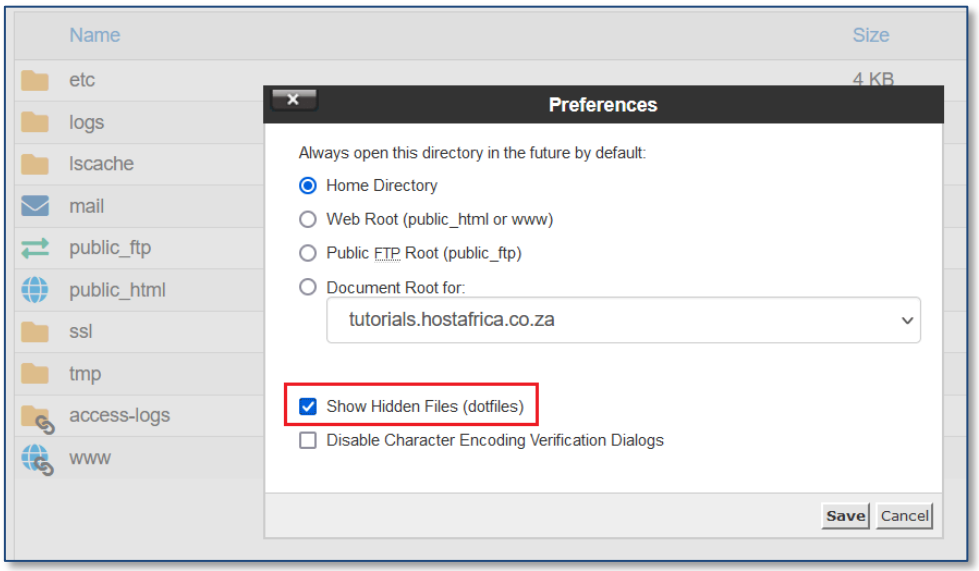
4. CPanel Apache:

Para realizar las configuraciones correspondientes primeramente se debe:

1. Iniciar sesión en cPanel
2. Hacer clic en el icono Administrador de archivos en Archivos

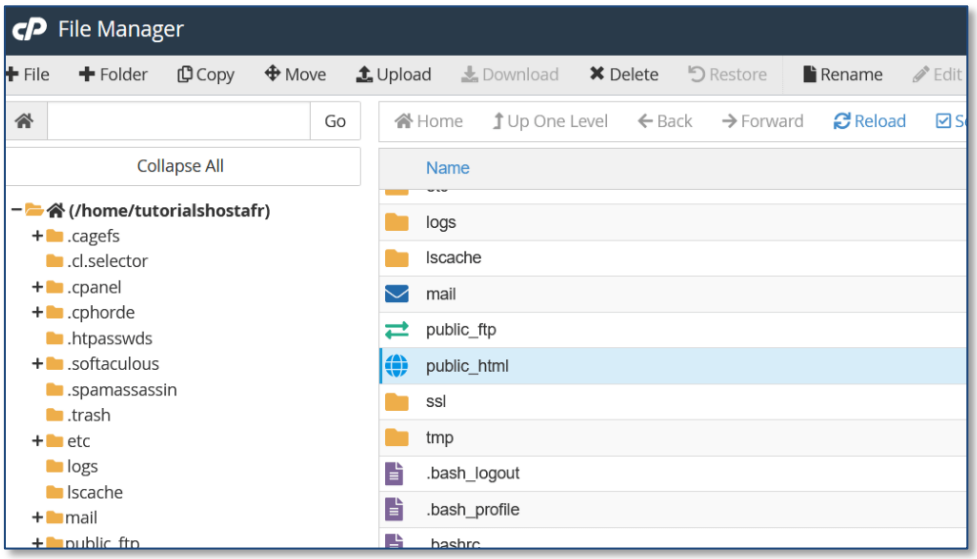


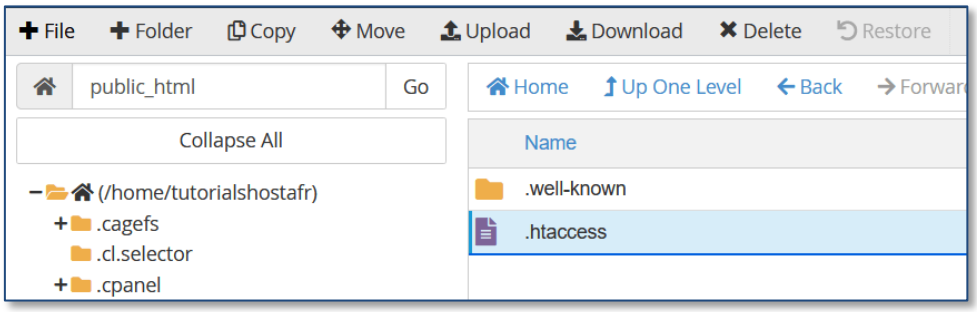
3. Hacer clic en el icono de **Configuración**, seleccionar **Mostrar archivos ocultos** (archivos de puntos) y hacer clic en Guardar.



4. Ir a la carpeta **public_html** y hacer doble clic en ella para buscar el archivo **.htaccess**.

El archivo **.htaccess** (HyperText Access o acceso de hipertexto) es un archivo de configuración del software de servidor Apache, que contiene las directivas que definen el comportamiento de este.



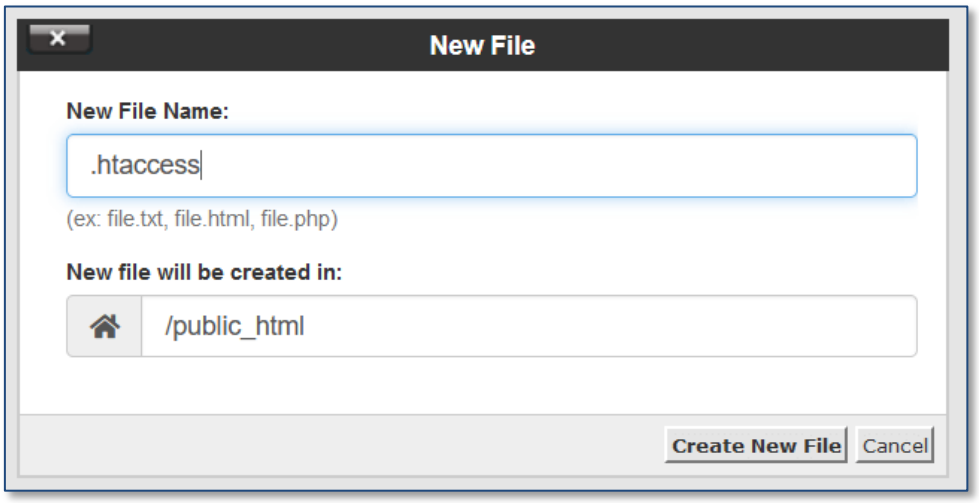


5. En caso de no contar con el archivo **.htaccess**, se debe crear el mismo, siguiendo los siguientes pasos:

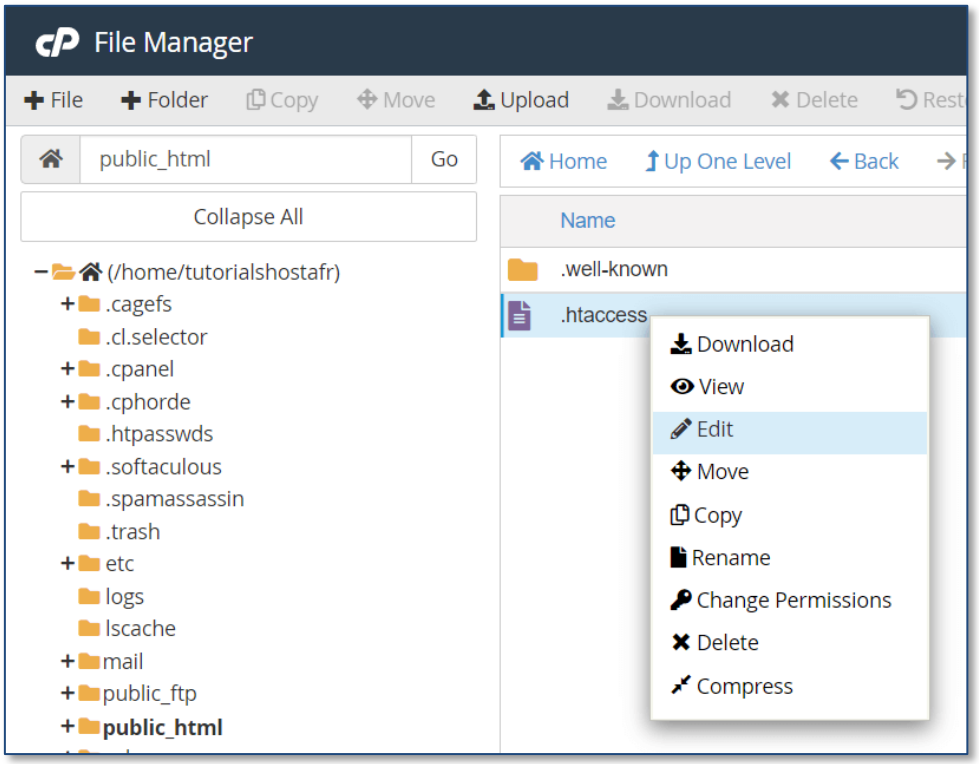
   5.1 Abrir la carpeta **public_html**.

   5.2 Una vez dentro de su carpeta **public_html**, haga clic en el icono Nuevo Archivo en la esquina superior izquierda.

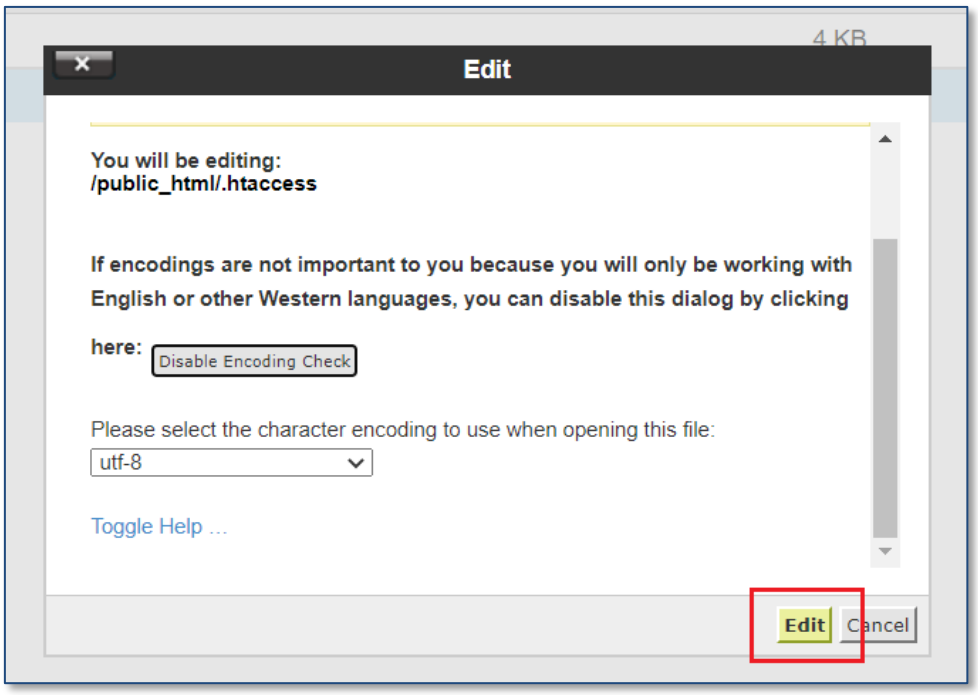
   5.3 Escriba **.htaccess** como nombre del archivo y haga clic en **Crear nuevo archivo**



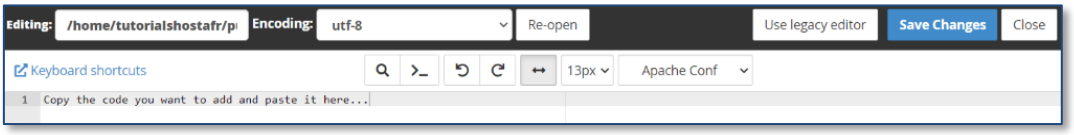
5.4 Una vez creado dicho archivo .**htaccess** o si ya cuenta con el mismo, agregar el código de configuración haciendo clic con el botón derecho sobre el mismo y seleccionar **Editar**.



5.5 Se deberá **habilitar/deshabilitar** la comprobación de codificación y seleccionar una codificación de caracteres para el archivo y hacer clic en **Editar**.



5.6 Abrir un editor de texto, copiar la configuración de seguridad de las cabeceras detallada a continuación y pegar dentro del espacio de texto.



Configuración de la cabecera X-Frame-Options:

```
<IfModule mod_headers.c>
	Header set X-Frame-Options "SAMEORIGIN"
</IfModule>
```

Configuración de la cabecera Strict-Transport-Security:

```
<IfModule mod_headers.c>
	Header set Strict-Transport-Security "max-age=31536000; includeSubDomains"
</IfModule>
```

Configuración de la cabecera X-XSS-Protection:

```
<IfModule mod_headers.c>
	Header set X-Xss-Protection "1; mode=block"
</IfModule>
```

Configuración de la cabecera Content-Security-Policy:

```
<IfModule mod_headers.c>
	Header set Content-Security-Policy "upgrade-insecure-requests"
</IfModule>
```

Configuración de la cabecera X-Content-Type-Options:

```
<IfModule mod_headers.c>
	Header set X-Content-Type-Options "nosniff"
</IfModule>
```

Configuración de la cabecera Cache-Control:

```
<IfModule mod_headers.c>
	Header set Cache-Control "private, no-cache, no-store, max-age=0”
</IfModule>
```

5. Servidor JBOSS:

Configuración de la cabecera X-Frame-Options:

Para realizar la configuración correspondiente, seguir los siguientes pasos:

1. Ir a la carpeta
   `<LISA_HOME>/IdentityAccessManager/standalone/configuration.`
2. Realizar una copia de seguridad de la versión independiente.xml.
3. Abrir independiente.xml de la carpeta de configuración.
4. Buscar el bloque

   `<subsystem xmlns="urn:jboss:domain:undertow:4.0">.`
   Tendrá un nombre de etiqueta `<host>` para (**10.6**)
   `<subsystem xmlns="urn:jboss:domain:undertow:12.0" >`
   bloquear y buscar la etiqueta `<host>` para (**10.7.x**)
5. En la etiqueta `<host >` y debajo de
   `<http-invoker security-realm="ApplicationRealm"/>`
   agregar la siguiente línea: `<filter-ref name="X-Frame-Options"/>.`

6. Ahora agregar la siguiente línea debajo de la etiqueta
   `<handlers></handlers>` en el mismo
   `<subsystem xmlns="urn:jboss:domain:undertow:4.0">block` para 10.6 o en
   `<subsystem xmlns="urn:jboss:domain:undertow:12.0">` para 10.7.x

```
<filters>
       <response-header name="X-Frame-Options" header-name="X-Frame-Options"
       header-value="SAMEORIGIN"/>
</filters>
</subsystem>
```

7. La sección se verá de la siguiente manera al realizar los cambios:
   `<subsystem xmlns="urn:jboss:domain:undertow:4.0">` o

```
<subsystem xmlns="urn:jboss:domain:undertow:4.0"> or   <subsystem
xmlns="urn:jboss:domain:undertow:12.0">
<buffer-cache name="default"/>
<server name="default-server">
<http-listener name="default" socket-binding="http" redirect-socket="https"
enable-http2="true"/>
<https-listener name="https" socket-binding="https" security-
realm="iamRealm" enable-http2="true"/>
<host name="default-host" alias="localhost">
<location name="/" handler="welcome-content"/>
<http-invoker security-realm="ApplicationRealm"/>
      <filter-ref name="X-Frame-Options"/>
</host>
</server>
<servlet-container name="default">
<jsp-config/>
<websockets/>
</servlet-container>
<handlers>
<file name="welcome-content" path="${jboss.home.dir}/welcome-content"/>
</handlers>
<filters>
      <response-header name="X-Frame-Options" header-name="X-Frame-Options"
      header-value="SAMEORIGIN"/>
</filters>
</subsystem>
```

8. Guardar y salir.

9. Reiniciar IAM.

10. Hacer que el equipo de seguridad vuelva a ejecutar el análisis y esto resolverá la vulnerabilidad.

Configuración de la cabecera Strict-Transport-Security:

Para su configuración, agregar los siguientes pasos:

1. Ir a la carpeta
   `<LISA_HOME>/IdentityAccessManager/standalone/configuration.`
2. Realizar una copia de seguridad de la versión independiente.xml.
3. Abrir independiente.xml de la carpeta de configuración.
4. Buscar el bloque
   `<subsystem xmlns="urn:jboss:domain:undertow:4.0">.`
   Tendrá un nombre de etiqueta `<host>` para (**10.6**)
   `<subsystem xmlns="urn:jboss:domain:undertow:12.0" >`
   bloquear y buscar la etiqueta `<host>` para (**10.7.x**)
5. En la etiqueta `<host >` y debajo de
   `<http-invoker security-realm="ApplicationRealm"/>`
   `<filter-ref name="strict-transport-security"/>`
6. Ahora agregar la siguiente línea debajo de la etiqueta
   `<handlers></handlers>`
   en el mismo
   `<subsystem xmlns="urn:jboss:domain:undertow:4.0">block`
   para 10.6 o en
   `<subsystem xmlns="urn:jboss:domain:undertow:12.0">` para 10.7.x

```
<filters>
      <filter-ref name="strict-transport-security"/>
</filters>
</subsystem>
```

7. La sección se verá de la siguiente manera al realizar los cambios:
   `<subsystem xmlns="urn:jboss:domain:undertow:4.0">`
   o

```
<subsystem xmlns="urn:jboss:domain:undertow:4.0"> or   <subsystem
xmlns="urn:jboss:domain:undertow:12.0">
<buffer-cache name="default"/>
<server name="default-server">
<http-listener name="default" socket-binding="http" redirect-socket="https"
enable-http2="true"/>
<https-listener name="https" socket-binding="https" security-
realm="iamRealm" enable-http2="true"/>
<host name="default-host" alias="localhost">
<location name="/" handler="welcome-content"/>
<http-invoker security-realm="ApplicationRealm"/>
       <filter-ref name="strict-transport-security"/>
</host>
</server>
<servlet-container name="default">
<jsp-config/>
<websockets/>
</servlet-container>
<handlers>
<file name="welcome-content" path="${jboss.home.dir}/welcome-content"/>
</handlers>
<filters>
       <response-header name="strict-transport-security" header-
       name="Strict-Transport-Security" header-value="max-age=31536000;
       includeSubDomains"/>
</filters>
```

8. Guardar y salir.

9. Reiniciar IAM.

10. Hacer que el equipo de seguridad vuelva a ejecutar el análisis y esto resuelve la vulnerabilidad.

Configuración de la cabecera X-XSS-Protection:

Para su configuración, agregar los siguientes pasos:

1. Ir a la carpeta
   `<LISA_HOME>/IdentityAccessManager/standalone/configuration.`
2. Realizar una copia de seguridad de la versión independiente.xml.

3. Abrir independiente.xml de la carpeta de configuración.
4. Buscar el bloque
   `<subsystem xmlns="urn:jboss:domain:undertow:4.0">.`
   Tendrá un nombre de etiqueta `<host>` para (**10.6**)
   `<subsystem xmlns="urn:jboss:domain:undertow:12.0" >`
   bloquear y buscar la etiqueta `<host>` para (**10.7.x**)
5. En la etiqueta `<host >` y debajo de
   `<http-invoker security-realm="ApplicationRealm"/>`
   `<filter-ref name="x-xss-protection"/>`
6. Ahora agregar la siguiente línea debajo de la etiqueta
   `<handlers></handlers>`
   en el mismo
   `<subsystem xmlns="urn:jboss:domain:undertow:4.0">block`
   para 10.6 o en
   `<subsystem xmlns="urn:jboss:domain:undertow:12.0">` para 10.7.x

```
<filters>
       <response-header name="x-xss-protection" header-name="X-
       XSS-Protection" header-value="1; mode=block"/>
</filters>
</subsystem>
```

7. La sección se verá de la siguiente manera al realizar los cambios:
   `<subsistema xmlns="urn:jboss:domain:undertow:4.0">`
   o

```
<subsystem xmlns="urn:jboss:domain:undertow:4.0"> or   <subsystem
xmlns="urn:jboss:domain:undertow:12.0">
<buffer-cache name="default"/>
<server name="default-server">
<http-listener name="default" socket-binding="http" redirect-socket="https"
enable-http2="true"/>
<https-listener name="https" socket-binding="https" security-realm="iamRealm"
enable-http2="true"/>
<host name="default-host" alias="localhost">
<location name="/" handler="welcome-content"/>
<http-invoker security-realm="ApplicationRealm"/>
       <filter-ref name="x-xss-protection"/></host>
</server>
<servlet-container name="default">
<jsp-config/>
<websockets/>
</servlet-container>
<handlers>
<file name="welcome-content" path="${jboss.home.dir}/welcome-content"/>
</handlers>
<filters>
       <response-header name="x-xss-protection" header-name="X-XSS-Protection"
       header-value="1; mode=block"/>
</filters>
</subsystem>
```

8. Guardar y salir.
9. Reiniciar IAM.
10. Hacer que el equipo de seguridad vuelva a ejecutar el análisis y esto resuelve la vulnerabilidad.

Configuración de la cabecera Content-Security-Policy:

Para su configuración, agregar los siguientes pasos:

1. Ir a la carpeta `<LISA_HOME>/IdentityAccessManager/standalone/configuration.`
2. Realizar una copia de seguridad de la versión independiente.xml.
3. Abrir independiente.xml de la carpeta de configuración.
4. Buscar el bloque `<subsystem xmlns="urn:jboss:domain:undertow:4.0">.`

Tendrá un nombre de etiqueta `<host>` para (**10.6**)
`<subsystem xmlns="urn:jboss:domain:undertow:12.0" >`
bloquear y buscar la etiqueta `<host>` para (**10.7.x**)

5. En la etiqueta `<host >` y debajo de
`<http-invoker security-realm="ApplicationRealm"/>`
`<filter-ref name="content-security-policy"/>`

6. Ahora agregar la siguiente línea debajo de la etiqueta
`<handlers></handlers>`
en el mismo
`<subsystem xmlns="urn:jboss:domain:undertow:4.0">block`
para 10.6 o en
`<subsystem xmlns="urn:jboss:domain:undertow:12.0">` para 10.7.x

```
<filters>
      <filter-ref name="content-security-policy"/>
</filters>
</subsystem>
```

7. La sección se verá de la siguiente manera al realizar los cambios:
`<subsystem xmlns="urn:jboss:domain:undertow:4.0">`
o

```
<subsystem xmlns="urn:jboss:domain:undertow:4.0"> or   <subsystem
xmlns="urn:jboss:domain:undertow:12.0">
<buffer-cache name="default"/>
<server name="default-server">
<http-listener name="default" socket-binding="http" redirect-socket="https"
enable-http2="true"/>
<https-listener name="https" socket-binding="https" security-realm="iamRealm"
enable-http2="true"/>
<host name="default-host" alias="localhost">
<location name="/" handler="welcome-content"/>
<http-invoker security-realm="ApplicationRealm"/>
      <filter-ref name="content-security-policy"/></host>
</server>
<servlet-container name="default">
<jsp-config/>
<websockets/>
</servlet-container>
<handlers>
<file name="welcome-content" path="${jboss.home.dir}/welcome-content"/>
</handlers>
<filters>
      <response-header name="content-security-policy" header-name="content-
      security-policy" header-value="default-src ; style-src 'unsafe-inline';
      script-src * 'unsafe-inline' 'unsafe-eval'; img-src * data: 'unsafe-
      inline';         connect-src * 'unsafe-inline'; frame-src *;"/>
      </filters>
</subsystem>
```

8. Guardar y salir.
9. Reiniciar IAM.
10. Hacer que el equipo de seguridad vuelva a ejecutar el análisis y esto resolverá la vulnerabilidad.

Configuración de la cabecera X-Content-Type-Options:

Para su configuración, agregar los siguientes pasos:

1. Ir a la carpeta `<LISA_HOME>/IdentityAccessManager/standalone/configuration.`
2. Realizar una copia de seguridad de la versión independiente.xml.
3. Abrir independiente.xml de la carpeta de configuración.

4. Buscar el bloque
`<subsystem xmlns="urn:jboss:domain:undertow:4.0">.`
Tendrá un nombre de etiqueta `<host>` para (**10.6**)
`<subsystem xmlns="urn:jboss:domain:undertow:12.0" >`
bloquear y buscar la etiqueta `<host>` para (**10.7.x**)

5. En la etiqueta `<host >` y debajo de
`<http-invoker security-realm="ApplicationRealm"/>`
`<filter-ref name="x-Content-type-options"/>`

6. Ahora agregar la siguiente línea debajo de la etiqueta
`<handlers></handlers>`
en el mismo
`<subsystem xmlns="urn:jboss:domain:undertow:4.0">block`
para 10.6 o en
`<subsystem xmlns="urn:jboss:domain:undertow:12.0">` para 10.7.x

```
<filters>
       <response-header name="x-Content-type-options" header-
       name="X-Content-Type-Options" header-value="nosniff"/>
</filters>
</subsystem>
```

7. La sección se verá de la siguiente manera al realizar los cambios:
`<subsystem xmlns="urn:jboss:domain:undertow:4.0">`
o

```
<subsystem xmlns="urn:jboss:domain:undertow:4.0"> or   <subsystem
xmlns="urn:jboss:domain:undertow:12.0">
<buffer-cache name="default"/>
<server name="default-server">
<http-listener name="default" socket-binding="http" redirect-socket="https"
enable-http2="true"/>
<https-listener name="https" socket-binding="https" security-realm="iamRealm"
enable-http2="true"/>
<host name="default-host" alias="localhost">
<location name="/" handler="welcome-content"/>
<http-invoker security-realm="ApplicationRealm"/>
      <filter-ref name="x-Content-type-options"/>
</server>
<servlet-container name="default">
<jsp-config/>
<websockets/>
</servlet-container>
<handlers>
<file name="welcome-content" path="${jboss.home.dir}/welcome-content"/>
</handlers>
<filters>
      <response-header name="x-Content-type-options" header-name="X-Content-
      Type-Options" header-value="nosniff"/></filters>
</subsystem>
```

8. Guardar y salir.
9. Reiniciar IAM.
10. Hacer que el equipo de seguridad vuelva a ejecutar el análisis y esto resolverá la vulnerabilidad.

Configuración de la cabecera Cache-Control:

Para su configuración, seguir los siguientes pasos:

1. Utilizar RESTEasy o utilizar ContainerResponseFilter; una interfaz de filtro proporcionada por JAX-RS.

2. Seguido escribir en el filtro personalizado el código fuente de la aplicación web.

   Esto se verá de la siguiente manera:

```
@Provider
public class YourCustomFilter implements ContainerResponseFilter{

// you can check the actual string value by using method "getStringHeaders" on 'resp'
below
private static final String CACHE_CONTROL = "cache-control";

@Override
public void filter(ContainerRequestContext req,
        ContainerResponseContext resp) throws IOException {

    if(resp.getHeaders().containsKey(CACHE_CONTROL)){
       resp.getHeaders().remove(CACHE_CONTROL);
       resp.getHeaders().add(CACHE_CONTROL, "no-transform, max-age=3600");
    }
    resp.getHeaders().add(CACHE_CONTROL, "no-transform, max-age=3600");

  }

}
```

6. Servidor Apache Tomcat:

Configuración de la cabecera X-Frame-Options:

Para su configuración, utilizar el siguiente procedimiento para habilitar la función antiClickJackingEnabled de Tomcat.

```
Header always append X-Frame-Options TRUE
```

Configuración de la cabecera Strict-Transport-Security:

Para su configuración, utilizar el siguiente procedimiento para habilitar la función hstsEnabled de Tomcat.

```
Header always append Strict-Transport-Security TRUE
```

Configuración de la cabecera X-XSS-Protection:

Para su configuración, utilizar el siguiente procedimiento para habilitar la función xssProtectionEnabled de Tomcat.

```
Header always append X-XSS-Protection: 1; mode=blocktrue
```

Configuración de la cabecera Content-Security-Policy:

Para su configuración, utilizar la siguiente clase web.xml, creada en Java:

```
<filter>
	<filter-name>ContentSecurityPolicy</filter-name>
	<filter-class>YourPackagePath.ContentSecurityPolicyFilter</filter-class>
</filter>
<filter-mapping>
	<filter-name>ContentSecurityPolicy</filter-name>
	<url-pattern>/*</url-pattern>
</filter-mapping>
```

Seguido se implementará el siguiente valor en la cabecera como resultado

```
default-src 'none'; style-src 'self' 'unsafe-inline'; script-src 'self' 'unsafe-inline'
'unsafe-eval'; img-src 'self'; frame-src 'self'; connect-src 'self'; form-action 'self';
reflected-xss block
```

Configuración de la cabecera X-Content-Type-Options:

Para su configuración, utilizar el siguiente procedimiento para habilitar la función blockContentTypeSniffingEnabled de Tomcat.

```
Header always append X-Content-Type-Options TRUE
```

Configuración de la cabecera Cache-Control:

Para su configuración, utilizar la siguiente clase web.xml, creada en Java: `filter`

```
    <filter-name>ExpiresFilter</filter-name>
    <filter-class>org.apache.catalina.filters.ExpiresFilter</filter-class>
    <init-param>
        <param-name>ExpiresDefault</param-name>
        <param-value>access plus 0 seconds</param-value>
    </init-param>
</filter>

<filter-mapping>
    <filter-name>ExpiresFilter</filter-name>
    <url-pattern>/*</url-pattern>
    <dispatcher>REQUEST</dispatcher>
</filter-mapping>
```

Referencias:

- https://www.cert.gov.py/wp-content/uploads/2022/07/Guia_de_Seguridad_-__HSTS_un_mecanismo_de_seguridad_adicional_a_HTTPs.pdf
- https://www.ryadel.com/en/http-security-headers-guide-tutorial-how-to-iis-apache-nginx/
- https://crashtest-security.com/enable-security-headers/
- https://www.babelgroup.com/es/Media/Blog/Enero-2020/cabeceras-seguridad-servidores
- https://knowledge.broadcom.com/external/article/201461/sv-vulnerability-xframeoptions-or-conte.html
- https://raddy.dev/blog/http-security-headers-apache-whm/
- https://tomcat.apache.org/tomcat-10.0-doc/config/filter.html#HTTP_Header_Security_Filter