



Content-Security-Policy

Decide con quién '*hablas*'

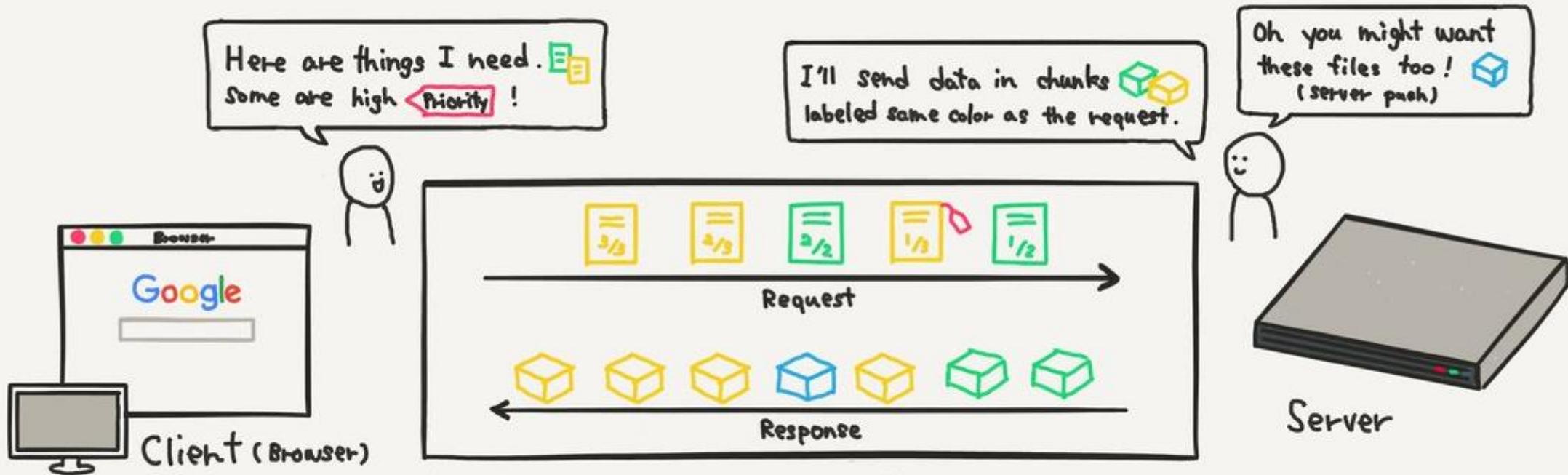
Si te vas a dedicar a Internet

has de conocer cómo funciona Internet (al menos un poco).

El CSP es complejo

así que hoy te voy a enseñar 'cuatro pinceladas'...

HTTP 2 NEW!



HTTP2 use one communication line per origin
TCP connection domain

It use Stream[⚡] to transfer data.

Ataques

3/4 de los ataques que recibimos se ejecutan en el servidor, así que ¿lo tienes al día?

- ¿hosting compartido?: pregunta a tu hosting
- ¿VPS o dedicado?: mantenlo al día.

El ataque que más se usa es el XSS:

Los *Cross-Site Scripting* (XSS) son un tipo de ataque en la que se inyectan scripts maliciosos en sitios web benignos y de confianza.

Victimas

Aunque pueda parecer que tu sitio es el afectado por un ataque XSS, en realidad el que lo sufre es tu usuario.

Estos sistemas XSS permiten, por ejemplo:

- Redirigir al usuario a otro sitio
- Instalarle malware
- Robo de identificadores / contraseñas
- ...

Animated Weather Widget

Este plugin incluía un iframe que cargaba un código de *CoinHive*, minando criptomonedas *Monero*.

A los usuarios se les saturaba el navegador mediante la minería.

+ info: wordfence.com



Animated weather widget by weatherfor.us

Weather for us widgets that get accurate and beautiful weather forecasts without worrying about any complicated configurations.

[Download Version 1.5](#)

Description Installation FAQ Screenshots Changelog Stats Support Reviews Developers

Requires: 3.4 or higher
Compatible up to: 3.9.11
Last Updated: 2 years ago
Active Installs: 4,000+

Ratings

Rating	Count
5 stars	9
4 stars	0
3 stars	0
2 stars	1
1 star	2

Skins

Currently Weather for us has 2 skins as we are working on more:

- *720px wide large weather widget* - that directly displays inline to place where you drop it. This widget is recommended for placing in your site or article footers. Excellent for embedding anywhere where you want a wide neat view and a clear focus on current weather conditions.
- *Mini preview mode weather widget* - that shows weather prediction for today with location and temperature. Clicking on the preview opens a beautiful semi-transparent background with live 720px wide widget in center of your page. This widget is fluid i.e. adjusts to width of your panel, but recommended for widths between 120px to 200px range. The weather icon, text & location automatically adjusts to your sidebar width of your panel. You don't need to worry about configuration mess at all!

Great Features include

- Present the weather in your sidebar anywhere on your site
- Weather available for nearly any location in the world
- Cool and familiar style

Author

Weatherfor.us

Donate to this plugin »

Support

Got something to say? Need help?

[View support forum](#)

Translations

Content Security Policy

- Es un estándar de la W3C
- Permite añadir cabeceras que hacen una “lista blanca” de sitios válidos.

Algo sencillo:

```
Content-Security-Policy: default-src 'self' https:;
```

Directivas de recuperación

- default-src**: es lo que cumplirá por defecto.
- font-src**: para fuentes.
- frame-src**: para iframes.
- img-src**: para imágenes.
- media-src**: para media (tag <video>, <audio>...)
- script-src**: para <script>
- style-src**: para hojas de estilo.
- ...

Ejemplos

```
Content-Security-Policy: font-src 'self'  
https://fonts.googleapis.com;
```

```
Content-Security-Policy: img-src 'self'  
https://example.cloudfront.net/;
```

```
Content-Security-Policy: script-src 'self'  
https://cdnjs.cloudflare.net https://code.jquery.com  
bootstrapcdn.com;
```

Mega Ejemplo

```
Content-Security-Policy: default-src 'self' https://; font-src 'self'  
https://fonts.googleapis.com; img-src 'self'  
http://example.cloudfront.net/; script-src 'self'  
http://cdnjs.cloudflare.net https://code.jquery.com  
bootstrapcdn.com;
```

En una línea se pueden añadir muchas fuentes de origen.

Atención: si usas directivas de estas, no puedes usar CSS o JavaScript inline...

Venga, pongamos cosas "inline"

```
Content-Security-Policy: default-src 'self' https://; script-src  
'self' 'unsafe-inline' 'unsafe-eval';
```

Si queremos añadir código inline, se puede, pero claro, cualquier "hackeo" que inyecte código funcionará y todo esto no servirá para nada.

¿Cómo aseguro un script local?

Mediante "nonce":

```
<script nonce='alertJavierCasares'>  
alert('Hola Javier Casares');  
</script>
```

Content-Security-Policy: script-src 'nonce-alertJavierCasares';

¿Cómo aseguro un script local?

Mediante "sha256":

```
<script>  
alert('Hola Javier Casares');  
</script>
```

Content-Security-Policy: script-src 'sha256-
CA28231318BF6923511C9AD992F0AF911F834132528DB44323D2A8E9D484CA2B' ;

Evita el aviso de elementos sin HTTPS

Es tan sencillo como:

Content-Security-Policy: upgrade-insecure-request;

A partir de ese momento todas las peticiones locales a HTTP pasan a ser automáticamente HTTPS.



¿Pasan cosas? Reporting

Crea una URL que recoja los reportes:

Content-Security-Policy: default-src https:; report-uri https://example.com/reportcsp

```
{"csp-report": {  
  "document-uri": "https://example.com",  
  "referrer": "",  
  "blocked-uri": "http://example.com",  
  "violated-directive": "default-src https:",  
  "original-policy": "default-src https:; report-uri https://example.com/reportcsp"  
}}
```

Código en el servidor web

Estas directivas son "headers" que se pueden añadir en cualquier servidor web:

Apache HTTPD:

```
Header set Content-Security-Policy: default-src https:;
```

nginx:

```
add_header Content-Security-Policy: default-src https:;
```

Código en el functions.php | plugin

Podemos añadir una función que añada las cabeceras:

```
add_action('send_headers', function() {
    header("Content-Security-Policy: default-src https:;");
});
```

Plugins

Security Headers

wordpress.org/plugins/security-headers/



Plugins

WP Content Security Plugin

wordpress.org/plugins/wp-content-security-policy/

Policy Editor Content Security Policy allows the following entries - one per line:

*	Allows Anything (try to avoid)
'none'	Allows nothing. The single quotes are required.
'self'	Allows from the same domain (volume and host) only. The single quotes are required.
'strict-source'	Allows use of inline source elements - scripts, fonts, etc. The single quotes are required.
'unsafe-eval'	Allows unsafe execution of evaluated javascript code. The single quotes are required.
'unsafe-hashes'	The trust explicitly given to a nonce present in the markup, by accompanying it with a nonce or a hash, shall be propagated to all the scripts loaded by their root script. The single quotes are required.
data:	Allows loading resources from data scheme. This is insecure ; an attacker can also inject arbitrary data: URIs. Use this sparingly and definitely not for scripts.
mediastream:	Allows media stream URIs to be used as a content source.
allow-sameorigin	Allows loading resource from file system
key-pair	Allows loading resource over a secure connection from any domain (block insecure content)
domain.example.com	Allows loading resource from the specific domain, any scheme
*.example.com	Allows loading resource from any subdomain of the specified domain
http://www.example.com	Allows loading resource from the specific domain and this scheme

WP CONTENT SECURITY POLICY

Default SPC: 'self'

Plugins

HTTP Headers

wordpress.org/plugins/http-headers/



Herramientas

Hay muchas, pero estas son quizá las más útiles:

csp-evaluator.withgoogle.com

securityheaders.com

report-uri.com/home/tools

www.telerik.com/fiddler

Comienza el resumen final

Comienza, pero no es el final...

Directivas

- Hay muchas de recuperación (y habrá más):
<https://www.w3.org/TR/CSP/#directives-fetch>
- También las hay a nivel página:
<https://www.w3.org/TR/CSP/#directives-document>
- Y de navegación, ¿quién llama a mi formulario?:
<https://www.w3.org/TR/CSP/#directives-navigation>

Recibe avisos de ataques

- Recuerda configurar el reporting:

<https://www.w3.org/TR/CSP/#reporting>

Un ejemplo real

En este caso: casares.blog

Mi código básico de WordPress

Está en modo 'Report-Only' así que puedes copiar y pegar sin problema:

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

Mi código

Activado en modo "Report-Only":

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;  
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'  
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action  
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;  
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src  
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

Mi código

Cargamos los HTTP locales como HTTPS, siempre:

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;  
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'  
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action  
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;  
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src  
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

Mi código

Ponemos el dominio base a tener en cuenta:

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;  
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'  
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action  
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;  
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src  
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

Mi código

Qué hacer por defecto:

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;  
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'  
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action  
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;  
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src  
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

Mi código

Qué URLs puede cargar 'AJAX':

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;  
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'  
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action  
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;  
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src  
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

Mi código

Desde dónde se pueden cargar las fuentes:

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;  
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'  
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action  
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;  
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src  
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

Mi código

Cómo se pueden llamar mis formularios:

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;  
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'  
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action  
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;  
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src  
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

Mi código

Quién puede incluirme como iframe / frame:

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;  
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'  
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action  
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;  
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src  
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

Mi código

Qué iframes / frames puedo incluir:

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;  
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'  
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action  
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;  
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src  
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

Mi código

Desde donde se pueden llamar imágenes:

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;  
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'  
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action  
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;  
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src  
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

Mi código

Desde donde se pueden cargar los 'manifest':

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;  
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'  
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action  
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;  
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src  
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

Mi código

Qué URLs pueden tener mis <audio> o <video>:

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;  
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'  
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action  
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;  
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src  
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

Mi código

Qué objetos externos puedo cargar:

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;  
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'  
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action  
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;  
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src  
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

Mi código

Cómo ejecuto mis <script>:

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;  
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'  
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action  
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;  
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src  
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

Mi código

Desde dónde cargo las hojas de estilo CSS:

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;  
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'  
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action  
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;  
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src  
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

Mi código

Qué 'workers' pueden ejecutarse:

```
Content-Security-Policy-Report-Only upgrade-insecure-requests;  
base-uri www.casares.blog; default-src 'self' data: 'unsafe-inline'  
'unsafe-eval'; connect-src 'self'; font-src 'self'; form-action  
'self'; frame-ancestors 'self'; frame-src 'self' *.wordpress.org;  
img-src 'self' data: *.gravatar.com; manifest-src 'self'; media-src  
'self'; object-src 'none'; script-src 'self' 'unsafe-inline'  
'unsafe-eval'; style-src 'self' 'unsafe-inline'; worker-src 'self';
```

La respuesta

Si cargo las herramientas del navegador (Consola), me dice esto:

Content Security Policy: This site (<https://www.cañares.blog>) has a Report-Only policy without a report URI. CSP will not block and cannot report violations of this policy.

Content Security Policy: The page's settings observed the loading of a resource at <https://fonts.googleapis.com/css?family=Fira+Mono:400,700|Fira+Sans:400,700> ("style-src"). A CSP report is being sent.

Content Security Policy: The page's settings observed the loading of a resource at <https://platform.twitter.com/widgets.js> ("script-src"). A CSP report is being sent.

Content Security Policy: The page's settings observed the loading of a resource at https://pbs.twimg.com/card_img/1028269421448445952/Ze3zGrti?format=jpg&name=144x144_2 ("img-src"). A CSP report is being sent.

Content Security Policy: The page's settings observed the loading of a resource at <https://platform.twitter.com/css/timeline.01c17c5f061c8031af501b58dbf35385.light.ltr.css> ("style-src"). A CSP report is being sent.

Content Security Policy: The page's settings observed the loading of a resource at <https://syndication.twitter.com/i/jot> ("form-action"). A CSP report is being sent.

Content Security Policy: The page's settings observed the loading of a resource at <https://syndication.twitter.com/i/jot> ("frame-src"). A CSP report is being sent.

Subresource Integrity

Bonus Track

Que no modifiquen tus scripts externos

Cuando llamas a scripts externos:

```
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
```

Aplica la integridad con un *hash*. Si cambia su contenido, no se cargará ni ejecutará.



Plugins

Subresource Integrity (SRI) Manager

wordpress.org/plugins/wp-sri/

Los deberes

esta vez, deberes para varios días...

Información / Herramientas

- W3C: [Content Security Policy](#)
- Google: [Content Security Policy](#)
- Mozilla: [Content Security Policy](#)
- CSP: [Quick Reference Guide](#)

- Report URI: [Generador de CSP](#)
- SRI [Hash Generator](#)

Por cierto

El CSP deja atrás otras cabeceras...

Cabeceras que ya son obsoletas

El CSP sustituye estos sistemas de seguridad:

```
X-Frame-Options "SAMEORIGIN";  
X-XSS-Protection "1; mode=block";  
X-Content-Type-Options "nosniff";
```

Si utilizas estos sistemas, sustitúyelos progresivamente por el CSP.

Un poco de humor...

¿atacamos alguna web con XSS?

Harlem Shake

- <http://www.lamoncloa.gob.es/>
- <http://www.rtve.es/>
- <http://www.elcorteingles.es/>
- <https://2018.valencia.wordcamp.org/>

Por cierto, ¿os habéis fijado que solo la web de la WordCamp tiene HTTPS?



Gracias

¿Preguntas?

www.javiercasares.com

javier @ casares .org