Content Security Policy (CSP)

Une *Content Security Policy (CSP)* ou stratégie de sécurité du contenu permet d'améliorer la sécurité des sites web en permettant de détecter et réduire certains types d'attaques, dont les attaques XSS (*Cross Site Scripting*) et les injections de contenu. Ces attaques peuvent être utilisées dans divers buts, comme le vol de données, le défacement de site ou la diffusion de *malware*.

CSP a été conçu pour être complètement rétro-compatible (à l'exception de la version 2 dans laquelle existent des incompatibilités décrites explicitement comme telles ; pour plus d'informations, se référer à la documentation du w3c (en anglais)). D'une part : les navigateurs qui ne prennent pas en charge le CSP fonctionnent parfaitement avec les serveurs qui l'implémentent et inversement. D'autre part, lorsque les sites ne fournissent pas les en-têtes correspondant, les navigateurs utilisent la règle de même origine (*same-origin policy*) pour les contenus.

Pour activer CSP, vous devez configurer vos serveurs web afin d'ajouter un en-tête (*header*) HTTP `Content-Security-Policy` aux réponses. Vous pouvez rencontrer des documents qui mentionnent `X-Content-Security-Policy` comme en-tête, il s'agit d'une version obsolète qu'il n'est plus utile de supporter.

Une autre possibilité consiste à utiliser l'élément HTML `<meta>` pour configurer la règle, par exemple : `<meta http-equiv="Content-Security-Policy" content="default-src 'self'; img-src https://*; child-src 'none';">`

Menaces

Réduction des attaques *cross site scripting* (XSS)

L'un des objectifs de CSP est la réduction et le rapport d'attaques XSS (injections de contenu). Les attaques XSS exploitent la confiance que les navigateurs ont dans le contenu reçu des serveurs. Des scripts malveillants peuvent être exécutés par le navigateur d'une victime parce que le navigateur fait confiance au serveur qui lui envoie des données même quand le contenu ne vient pas de là où il semble venir.

CSP permet aux administrateurs système de réduire ou éliminer les moyens de réaliser des attaques XSS en permettant de spécifier les domaines autorisés à fournir des scripts pour la page visitée. Un navigateur compatible avec CSP n'exécute que les scripts provenant d'une origine autorisée par les règles CSP reçues et ignore ceux qui ne sont pas autorisés. On peut ainsi bloquer les domaines non autorisés, les scripts *inline* (inclus dans une page HTML) ou associés à des événements via les attributs HTML dédiés.

Pour un niveau de protection le plus élevé possible, un site qui voudrait qu'aucun script ne puisse être exécuté peut désactiver tout simplement l'exécution de tout script.

Empêcher les écoutes du trafic

En plus de restreindre les domaines à partir desquels le contenu peut être chargé, le serveur peut indiquer quels protocoles doivent être utilisés et par exemple forcer l'utilisation de HTTPS afin d'améliorer la sécurité. Une stratégie de sécurité complète pour la transmission des données peut non seulement forcer l'utilisation de TLS via HTTPS mais aussi forcer l'utilisation de cookies sécurisés (qui ne peuvent être envoyés qu'en HTTPS) et indiquer de convertir automatiquement toutes les requêtes qui auraient été faites en HTTP simple en requêtes HTTPS. L'utilisation de l'en-tête `Strict-Transport-Security` permet de s'assurer que les navigateurs utilisent obligatoirement des connexions chiffrées en TLS (HTTPS).

Utiliser CSP

Configurer une stratégie CSP nécessite d'utiliser un en-tête HTTP `Content-Security-Policy` pour une page web et de spécifier une valeur pour contrôler les ressources que le navigateur est autorisé à charger pour cette page. Ainsi, une page qui charge et affiche des images peut autoriser les images stockées n'importe où mais n'autoriser les envois de formulaires que vers certaines adresses.

Créer votre règle CSP

On peut utiliser l'en-tête HTTP `Content-Security-Policy` pour définir la règle ainsi :

```
Content-Security-Policy: règle
```

La `règle` est une chaîne de caractères contenant la liste des règles qui constituent la règle CSP.

Écrire une règle

Une règle est définie par une série de directives qui décrivent chacune le comportement attendu pour un certain type de contenu ou pour l'ensemble des requêtes. Une règle peut inclure une directive `default-src` pour la règle par défaut qui s'applique aux ressources pour lesquelles aucune règle n'est définie. Pour les autres types de règle, on pourra se référer à la page `default-src`. Pour bloquer les scripts intégrés au code HTML (JavaScript *inline*) et l'utilisation de `eval()`, une règle doit au moins contenir une directive `default-src` ou `script-src`. Pour bloquer les modifications de style intégrées au code HTML (CSS *inline* avec les attributs HTML `<style>`) et l'utilisation des balises `style`, une règle doit au moins contenir une directive `default-src` ou `style-src`.

Exemples pour les cas courants

Cette section propose des règles CSP pour les scenarios les plus classiques.

Exemple 1

Ici, on souhaite que tout le contenu du site soit fourni par la même origine (on exclut les sous-domaines) :

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

Exemple 2

Pour un site dont tout le contenu est fourni par le site lui-même ou par les sous-domaines de `source-sure.example.net` (qui peut être un autre site) :

```
Content-Security-Policy: default-src 'self' *.source-sure.example.net
```

Exemple 3

Pour un site dont les images peuvent venir de n'importe où, les musiques et vidéos de `toto.local` ou `tata.local`, les scripts par `scripts.local` :

```
Content-Security-Policy: default-src 'self'; img-src *; media-src
toto.local tata.local; script-src scripts.local
```

Ici, les contenus doivent par défaut venir de la même origine que la page avec les exceptions précédemment décrites. Cela peut permettre aux utilisateurs d'afficher des images quelconques, mais de ne faire confiance qu'à certains domaines pour les musiques, vidéos et scripts.

Exemple 4

Pour un site dont les données sont critiques/privées et pour lequel toutes les données devraient être transmises en HTTPS depuis un domaine précis :

```
Content-Security-Policy: default-src https://confidentiel.example.net
```

Cette règle force l'utilisation de HTTPS et exclut tout usage de contenu ne venant pas de `https://confidentiel.example.net`.

Exemple 5

Pour un webmail qui permet d'afficher des mails incluant de l'HTML, des images provenant de n'importe où mais pas de JavaScript ou d'autres contenus potentiellement dangereux :

```
Content-Security-Policy: default-src 'self'; img-src *; child-src: *
```

On notera que dans cet exemple, on n'a pas de directive `script-src`. C'est la directive `default-src` qui indique le comportement par défaut et donc qui limite le chargement des scripts à l'origine.

Tester une règle CSP

Pour faciliter le déploiement de CSP, on peut configurer le serveur afin de rapporter uniquement les violations de règle sans appliquer réellement la règle. Ainsi, on peut s'assurer que la règle ne bloque pas les usages du site en récupérant les rapports de violation de la règle en test. On peut aussi tester des modifications d'une règle en place via ce même mécanisme.

Pour cela, il suffit d'utiliser l'en-tête `Content-Security-Policy-Report-Only`, comme cela :

```
Content-Security-Policy-Report-Only: règle
```

Si les en-têtes HTTP `Content-Security-Policy-Report-Only` et `Content-Security-Policy` sont tous deux présents dans la réponse du serveur, les deux règles seront respectées, ce qui permet le test d'une nouvelle règle quand il y en a déjà une en place.

La règle indiquée par `Content-Security-Policy` est appliquée tandis que celle fourni par `Content-Security-Policy-Report-Only` génère des rapports mais n'est pas appliquée.

Si une règle contient une directive `report-uri` valide, les navigateurs qui prennent en charge CSP doivent envoyer un rapport pour chaque violation de la règle qu'ils détectent.

Gérer les rapports

Par défaut, les violations de la règle de sécurité ne sont pas rapportées. Pour avoir des rapports de violation, il faut fournir directive `report-uri` avec au moins une URL valide à laquelle envoyer les rapports :

```
Content-Security-Policy: default-src 'self'; report-uri
http://reportcollector.example.com/collector.cgi
```

Il faut également configurer le serveur qui doit recevoir les rapports pour traiter les rapports en question et par exemple les stocker afin de les consulter.

Syntaxe des rapports de violation

Le rapport est un objet JSON qui contient :

`blocked-uri`
: L'URI de la ressource dont le chargement a été bloqué à cause du CSP. Si l'URI bloqué provient d'une origine différente de celle indiquée via `document-uri`, l'URI bloqué est tronqué et ne contient que le schéma, l'hôte et le port.

`disposition`
: La chaîne `"report"` si l'en-tête `Content-Security-Policy-Report-Only` a été utilisée ou `"enforce"` si `Content-Security-Policy` a été utilisée.

`document-uri`
: L'URI du document pour lequel la violation a eu lieu.

`effective-directive`
: La directive dont le non-respect a entraîné la violation.

`original-policy`
: La règle telle qu'indiquée dans l'en-tête HTTP `Content-Security-Policy`.

`referrer`
: Le *referrer* du document pour lequel la violation a eu lieu.

`script-sample`
: Les 40 premiers caractères du script, du gestionnaire d'évènement ou du style qui a entraîné la violation.

`status-code`
: Le code de statut HTTP de la ressource sur laquelle l'objet global a été instancié.

`violated-directive`
: Le nom de la directive, dans la règle, qui n'a pas été respectée.

Exemple de rapport de violation de règle

Si l'on considère une page `http://example.com/connexion.html`, qui utilise la règle CSP suivante (qui interdit tout par défaut et autorise les feuilles de style CSS provenant de `cdn.example.com`) :

```
Content-Security-Policy: default-src 'none'; style-src
cdn.example.com; report-uri /_/csp-reports
```

et qui contient le code HTML suivant :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Connectez-vous</title>
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    ... Contenu ...
  </body>
</html>
```

Dans cette situation, les clients qui visiteraient cette page la verrait avec les styles de base de leur navigateur car les feuilles de style autorisées ne peuvent venir que de `cdn.example.com` et non du site lui-même (l'origine même de la page) comme `<link rel="stylesheet" href="css/style.css">` l'indique au navigateur. En outre, un navigateur (qui supporte CSP) enverrait le rapport de violation de règle CSP suivant à l'adresse `http://example.com/_/csp-reports` à chaque visite de la page dont il est question :

```
{
  "csp-report": {
    "document-uri": "http://example.com/connexion.html",
    "referrer": "",
    "blocked-uri": "http://example.com/css/style.css",
    "violated-directive": "style-src cdn.example.com",
    "original-policy": "default-src 'none'; style-src cdn.example.c
  }
}
```

Comme vous pouvez le constater, le rapport inclus l'URI complète de la ressource dans `blocked-uri`. Ce n'est le cas en général. Ainsi, si la page avait essayé de charger la feuille de style `http://anothercdn.example.com/stylesheet.css`, le navigateur aurait indiqué seulement `"blocked-uri": "http://anothercdn.example.com/"`, c'est à dire l'origine et non l'URI complète car l'origine de la feuille bloquée est différente de l'origine du site lui-même. La spécification de la CSP, disponible en anglais sur le site du W3C, explique les raisons de ce comportement qui peut surprendre de prime abord. En résumé, ce comportement évite les risques de diffuser des informations confidentielles qui pourraient être incluses dans les URI des ressources provenant d'autres origines.

Compatibilité des navigateurs

Update compatibility data on GitHub

`Content-Security-Policy`	
Chrome	25

Edge	14
Firefox	23
IE	10 *
Opera	15
Safari	7
WebView Android	Oui
Chrome Android	Oui
Firefox Android	23
Opera Android	Oui
Safari iOS	7
Samsung Internet Android	Oui

What are we missing?

.. Support complet

* Voir les notes d'implémentation.

Cette fonctionnalité utilise un nom non-standard.

Il existe une incompatibilité spécifique dans certaines versions de Safari : si un en-tête `Content-Security-Policy` est défini mais qu'il n'y a pas d'en-tête `Same-Origin` , le navigateur bloquera le contenu du site courant et celui de l'extérieur en indiquant que la stratégie ne permet pas d'avoir ce contenu.

Voir aussi

- `Content-Security-Policy`
- `Content-Security-Policy-Report-Only`
- L'utilisation de CSP pour les WebExtensions.

- La gestion de CSP dans les web workers

Dernière modification : 21 nov. 2019, par des contributeurs MDN

Sujets associés

HTTP

Guides:

- Resources and URIs
- HTTP guide
- HTTP security

HTTP access control (CORS)

HTTP authentication

HTTP caching

HTTP compression

HTTP conditional requests

HTTP content negotiation

HTTP cookies

HTTP range requests

HTTP redirects

HTTP specifications

Feature policy

References:

- HTTP headers

- HTTP request methods
- HTTP response status codes
- CSP directives
- CORS errors
- Feature-Policy directives