

Résumé des attaques XSS

Définition

Une **XSS** (*Cross-Site Scripting*) est une faille qui permet à du contenu non fiable d'être interprété comme du code par le navigateur, dans le contexte d'un site vulnérable.

Elle survient souvent lorsqu'une application affiche une donnée utilisateur sans l'échapper ni la traiter selon son contexte.

Principaux types

- **XSS réfléchi** : une donnée reçue dans une requête est réaffichée immédiatement dans la page.
- **XSS stockée** : une donnée est enregistrée — par exemple un commentaire — puis affichée à d'autres utilisateurs.
- **XSS basée sur le DOM** : du code côté navigateur transmet une donnée non fiable à une API qui l'interprète comme du HTML ou du code.

Exemple simple

Une insertion avec `innerHTML` peut interpréter la valeur comme du HTML :

```
// À éviter avec une donnée non fiable  
element.innerHTML = valeurUtilisateur;
```

Si la donnée doit être affichée comme du texte, préférer `textContent` :

```
// Recommandé pour afficher du texte  
element.textContent = valeurUtilisateur;
```

Risques

Une XSS peut notamment permettre de modifier le contenu affiché, de tromper un utilisateur ou d'effectuer certaines actions avec ses droits. L'impact dépend des fonctionnalités et des protections de l'application.

Prévention

- **Échapper les données à l'affichage**, selon leur contexte : HTML, attribut, URL, etc.
- Utiliser l'échappement automatique fourni par les moteurs de templates ou frameworks.
- Éviter d'insérer des données non fiables avec `innerHTML` ou d'autres API qui interprètent des chaînes.
- Si l'application doit autoriser du HTML, utiliser une bibliothèque de **sanitisation spécialisée**, maintenue et configurée de façon restrictive.

- Ajouter une **Content Security Policy (CSP)** comme défense complémentaire.
- Protéger les cookies de session avec les attributs adaptés, comme `HttpOnly` et `Secure`.

Erreurs à éviter

- Se fier uniquement à une liste noire de caractères ou de balises.
- Croire que la validation des données remplace l'encodage à l'affichage.
- Nettoyer du HTML avec des expressions régulières ou du code maison.
- Considérer CSP, `HttpOnly` ou un pare-feu applicatif comme un remplacement à la correction de la faille.

À retenir



Traiter toute donnée externe comme non fiable.

- Pour afficher du texte, utiliser une méthode qui l'insère comme texte ;
- pour autoriser du HTML, recourir à une sanitisation spécialisée.

From:

<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:

<http://slamwiki2.kobject.net/sio/bloc3/attaques/xss?rev=1791154681>

Last update: **2026/10/05 00:58**

