

CROSS SITE SCRIPTING (XSS) ATTAQUES

Attaques par injection



SUP'MANAGEMENT
Réseau
Université Intercontinentale Libre



- **Fané Ousmane**

CEH, CCNA3, CCNA2, CCNA1...

Développeur Logiciel à Sup'Management

Promoteur de Donitech

Master 1 en Ingénierie des Réseaux Sécurité et Télécoms (IRST)

Licence en Réseaux et télécoms



CLAUSE DE NON-RESPONSABILITÉ

Tous les points de vue ou opinions présentés dans cette présentation sont uniquement les miens et ne représentent pas nécessairement mon employeur.

Je ne suis pas avocat et je ne vous donne pas de conseils juridiques

Je ne vous donne pas la permission ni ne vous autorise à faire quoi que ce soit.

En fait, ne faites jamais rien.





XSS

Cross

Site

Scripting



Sommaire

- Qu'est-ce que le Cross-Site Scripting ?
- Différents types de Cross-Site Scripting
- Impact du Cross-Site Scripting
- Moyens d'identifier les vulnérabilités XSS
- Prévention des attaques de Cross-Site Scripting



1- Qu'est-ce que XSS:

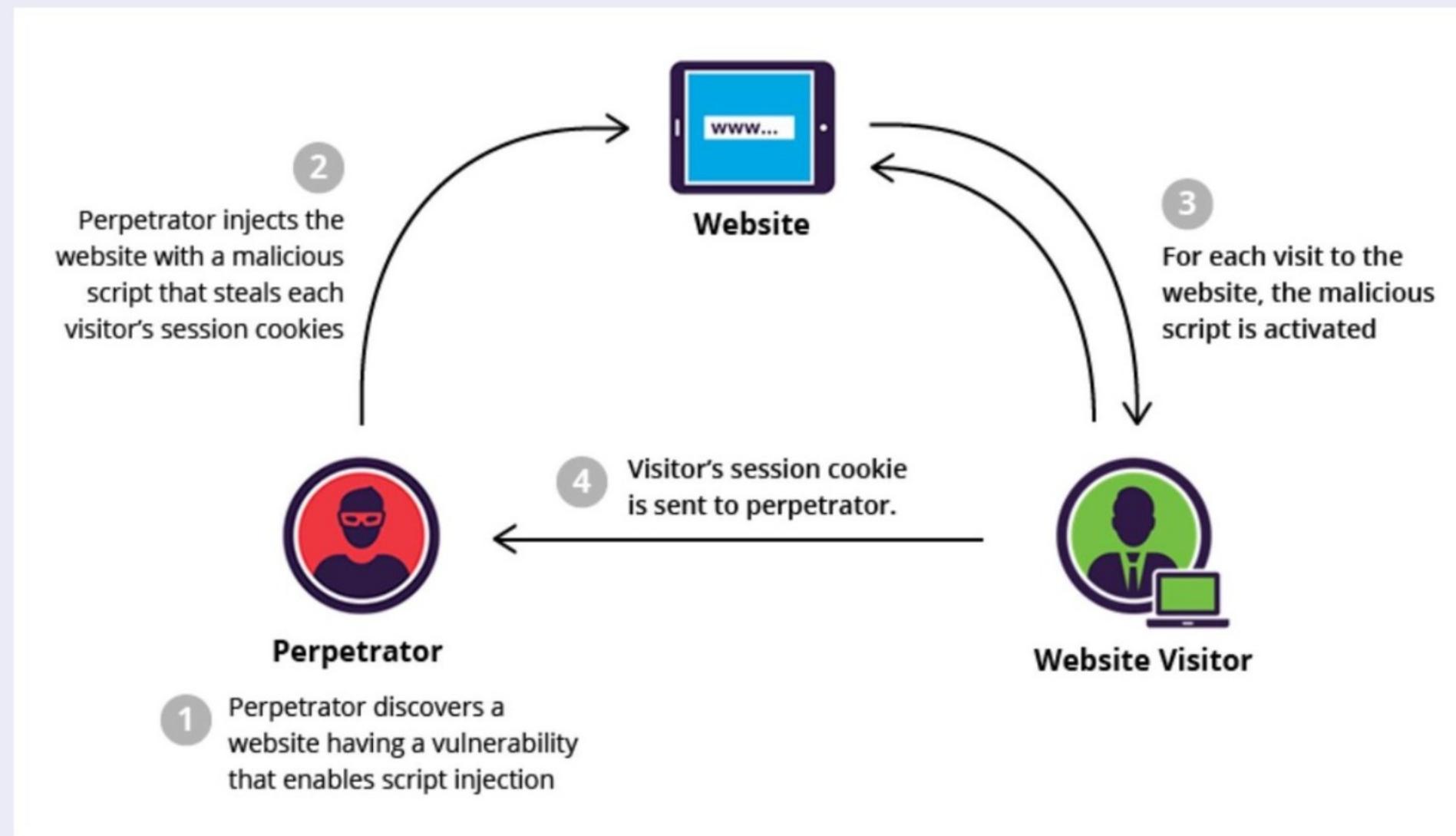
Le cross site scripting (XSS) est un vecteur d'attaque courant qui injecte du code malveillant dans une application Web vulnérable.

XSS diffère des autres vecteurs d'attaque Web (par exemple, les injections SQL) dans le sens où il ne cible pas directement l'application elle-même. Au lieu de cela, ce sont les utilisateurs de l'application Web qui sont à risque.

Une attaque réussie de type cross site scripting peut avoir des conséquences dévastatrices sur la réputation d'une entreprise en ligne et sur ses relations avec ses clients.



1- Qu'est-ce que XSS:





2- Types de XSS:

Il existe principalement trois types différents de vulnérabilité de cross-site scripting :

- **XSS réfléchi**

Une vulnérabilité XSS reflétée se produit lorsque la saisie d'un utilisateur à partir d'une URL ou de données POST est reflétée sur la page sans être stockée.

- **XSS persistant ou stocké**

Les vulnérabilités de script intersites stockées se produisent lorsque la charge utile est enregistrée, par exemple dans une base de données, puis est exécutée lorsqu'un utilisateur ouvre la page.

Les scripts intersites stockés sont très dangereux pour plusieurs raisons

- **XSS basé sur DOM**

La vulnérabilité XSS basée sur DOM se produit dans le DOM (Document Object Model) au lieu d'une partie du HTML.



2- Types de XSS:

Pendant des années, la plupart des gens ont considéré ces éléments (Stored, Reflected, DOM) comme trois types différents de XSS, mais en réalité, ils se chevauchent. Vous pouvez avoir à la fois du XSS basé sur le DOM stocké et réfléchi.

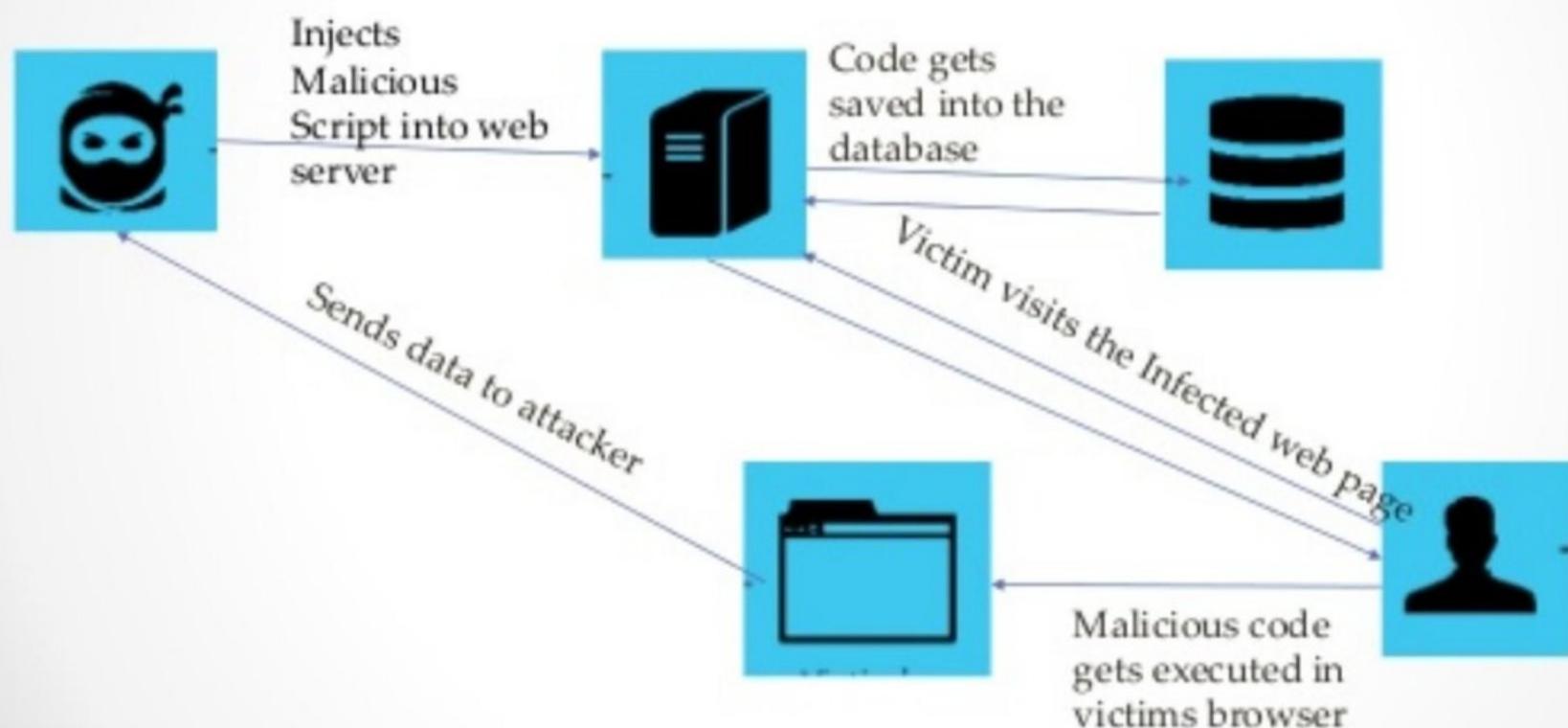
Vous pouvez également avoir des XSS non-DOM stockés et réfléchis, mais cela prête à confusion, donc pour aider à clarifier les choses, à partir de la mi-2012 environ, la communauté de recherche a proposé et commencé à utiliser deux nouveaux termes pour aider à organiser les types de XSS qui peuvent se produire :

1. Serveur XSS
2. Client XSS



Serveur XSS

How stored XSS is exploited





XSS client





3- Impact du XSS:

L'impact d'une vulnérabilité XSS exploitée varie beaucoup. Cela va de

- Redirection
- Détournement de session
- Contrefaçon de demande intersite
- Enregistrement de frappe
- Hameçonnage

En exploitant une vulnérabilité de script scripting un attaquant peut usurper l'identité de la victime et prendre le contrôle du compte. Si la victime dispose de droits d'administrateur, cela peut même conduire à l'exécution de code sur le serveur, selon l'application et les priviléges du compte.



4- Façons d'identifier et de vérifier les vulnérabilités XSS:

Les vulnérabilités de cross-site scripting peuvent être identifiées de 2 manières, à savoir :

- Analyse statique (révision du code source)
- Analyse dynamique (Fuzzing)

Outils d'analyse statique

- OWASP WAP - Projet de protection des applications Web
- RIPS - Un analyseur de code source statique
- Codacy :révisions et analyses de code automatisées

Outils d'analyse dynamique

- Burp suite
- Module complémentaire Hack bar Firefox ou module complémentaire burp
- Scanner de vulnérabilité automatisé (par exemple Arachni)



La démo de Préparez-vous commence



Tout le monde s'intéresse à quelque chose



5- Prévenir les scripts scripting

La prévention?

- Ne faites jamais confiance aux entrées de l'utilisateur
- Ne faites jamais confiance aux entrées de l'utilisateur
- Ne faites jamais confiance aux entrées de l'utilisateur
- Ne faites jamais confiance aux entrées de l'utilisateur
- Ne faites jamais confiance aux entrées de l'utilisateur
- Ne faites jamais confiance aux entrées de l'utilisateur
- Ne faites jamais confiance aux entrées de l'utilisateur
- Ne faites jamais confiance aux entrées de l'utilisateur



5- Prévenir les scripts scripting

Rappelons qu'une attaque XSS est un type d'injection de code : les entrées de l'utilisateur sont interprétées à tort comme du code de programme malveillant. Afin d'éviter ce type d'injection de code, une gestion sécurisée des entrées est nécessaire. Pour un développeur Web, il existe deux manières fondamentalement différentes d'effectuer une gestion sécurisée des entrées :

- Encodage, qui échappe à la saisie de l'utilisateur afin que le navigateur l'interprète uniquement sous forme de données, pas sous forme de code.
- Validation, qui filtre les entrées de l'utilisateur afin que le navigateur les interprète comme du code sans commandes malveillantes.



Prévenir XSS - Encodage

L'encodage consiste à échapper aux entrées de l'utilisateur afin que le navigateur les interprète uniquement comme des données et non comme du code. Le pseudocode suivant est un exemple de la façon dont les entrées utilisateur peuvent être codées à l'aide de l'échappement HTML.

```
print "<html>"  
print "Latest comment: "  
print encodeHtml(userInput)  
print "</html>"
```

Si l'entrée utilisateur était la chaîne <script>...</script>, le HTML résultant serait le suivant

```
<html>  
Latest comment:  
&lt;script&gt;...&lt;/script&gt;  
</html>
```



Prévenir XSS - Validation

La validation consiste à filtrer les entrées de l'utilisateur afin que toutes les parties malveillantes soient supprimées, sans nécessairement supprimer tout le code qu'elles contiennent. L'un des types de validation les plus reconnaissables dans le développement Web consiste à autoriser certains éléments HTML (tels que `` et ``) mais à en interdire d'autres (tels que `<script>`).

Il existe deux caractéristiques principales de la validation qui diffèrent selon les implémentations :

Stratégie de classification : les entrées utilisateur peuvent être classées à l'aide d'une liste noire ou d'une liste blanche.

Résultat de la validation : les entrées utilisateur identifiées comme malveillantes peuvent être rejetées ou nettoyées.



XSS n'est pas le problème de l'utilisateur comme n'importe quelle autre vulnérabilité de sécurité. Si cela affecte vos utilisateurs, cela vous affecte

J'espère que vous avez trouvé cette présentation utile

Les références

<https://www.netsparker.com>

<https://www.acunetix.com>

<https://excess-xss.com/>

<https://www.incapsula.com>

<https://www.owasp.org>

<https://www.google.com>



Merci questions et réponses

Courriel : ousmanefane08@gmail.com