Cross-Site Scripting (XSS)

|  |  |
| --- | --- |
| Elément | Description |
| **Compétences** | Repérer des vulnérabilités dans un code et les corriger. |
| **Objectifs** | Mettre en place les bonnes pratiques pour se prémunir d’une attaque XSS |
| **Durée estimée** | 60 min |
| **Répertoire de travail** | C:\Docker |
| **Fichiers sources** | E-183-ALL-XSS-http-payload.zip  E-183-ALL-XSS-stored.zip |
| **A produire** | Répondre aux questions directement dans ce document |
| **Moyens d’aide** |  |

# Informations générales

Le but de cet exercice est de vous mettre face à un code ayant des failles de sécurité (vulnérable face à XSS).

Cet exercice contient deux parties :

* Faille XSS avec surcharge http
* Faille XSS stockée

Il y aura donc deux environnement Docker à déployer :

* docker-xss-http-payload.zip
* docker-xss-stored.zip

# Questions

## Définissez avec vos propres mots en quoi consiste l’échappement de caractères et quel est son but

C’est un caractère qu’on peut nous même définir comment caractère d’échappement pour dire qu’il y a un espace ou un saut de ligne.

## Donner quelques caractères courants à échapper (ASCII + UTF8)

A l’aide de la table ASCII suivante, donnez 6 caractères courants à échapper.

A table of numbers and symbols

Description automatically generated

Vous noterez le symbole et le caractère échappé (en y incluant le symbole d’échappement) :

0x20 => Espace

0x2F = /

0x3B = ;

0x5C = \

0x25 = %

# XSS (HTTP payload)

## Contexte

L’exercice 1 est une fonction de recherche (**http://localhost:8087/search-results?q=mot\_clé)** qui permet à vos visiteurs de faire une recherche.

## Repérer dans le code les vulnérabilités

Dans le code fourni, se trouvent un élément qui permettent à un attaquant d’exploiter votre site.

Quels est cet élément :

Dans la balise <script>, searchTerm, sans aucune validation ou échappement.

## Exploiter la vulnérabilité

Essayez d’exploiter l’URL pour y insérer un script. Vous serez ainsi dans le rôle de l’attaquant qui essaie de forger un lien pour attaquer une cible.

*Indice : essayer de mettre de balises <script>.*

*Expliquez comment fonctionne la faille :*

*http://localhost:8087/search-results?q=%3Cscript%3Ealert(%27XSS%27)%3C/script%3E*

## Corriger la vulnérabilité

Mettez en place un élément dans votre code qui empêchera un attaquant d’exploiter les vulnérabilités de votre site.

Qu’avez-vous mis en place ?

# XSS (Stored)

## Contexte

L’exercice 2 est un livre d’or qui permet à vos visiteurs de donner leur avis sur votre entreprise.

## Repérer dans le code les vulnérabilités

Dans le code fourni, se trouvent deux éléments qui permettent à un attaquant d’exploiter votre site.

Quels sont ces éléments :

## Exploiter la vulnérabilité

Essayez d’exploiter le formulaire pour y insérer un script ou un lien affectant les autres utilisateurs du site.

*Indice : essayer de mettre de balises.*

Expliquez comment fonctionne la faille :

## Exploiter la vulnérabilité 2

Vous avez remarqué qu’un attaquant a pu manipuler votre site à l’aide du commentaire :

<img src="x" onerror="alert('XSS')">

Expliquez ce que l’attaquant à chercher à provoquer et quel méchanisme a-t-il voulu contourner.

## Corriger la vulnérabilité

Mettez en place deux éléments dans le code qui empêcheront un attaquant d’exploiter les vulnérabilités de votre site.

Qu’avez-vous mis en place ?