

## Web 2.0 Hacking : Introduction

Écrit par The\_NeTpSyChO

Dimanche, 03 Février 2008 12:53 - Mis à jour Mardi, 05 Février 2008 10:28

---

Voilà le premier d'une longue série d'articles sur la sécurité des applications web 2.0, nous verrons le coté offensif et défensif. Vous verrez des fois les phrases sont tournées de manière assez : spéciale, à l'origine certains de ces articles devaient sortir sur DG-SC e-zine mais il semblerait que ce soit retardé donc je préfère les publier petit à petit ici ... au mieux je publierais une suite dans DG-SC et je mettrais un lien vers le mag sur mon site :).

Bon trêve de bla bla : aujourd'hui une petite introduction pour vous mettre en jambe, et surtout vous mettre l'eau à la bouche !

let's go :

### Introduction :

En nos temps troublés nous voyons apparaitre de plus en plus de sites dit Web 2.0. Mais que cache ce nom qui sonne si bien, ou pas. Ben au fait il n'y a que du vieux que l'on a remixé afin de faire du neuf. Mais comme on dit par chez moi c'est dans les vieux pots que l'on fait les meilleurs soupes, sauf que si le pot est pourri la soupe sera immangeable !

Nous y voilà : AJAX Asynchronous Javascript And XML (ou AJAJ avec JSON), c'est ça la soupe toute pourrie ! Je vais donc essayer dans cet article (enfin dans cette série d'articles) de vous montrer qu'il est possible de mettre en place facilement des attaques de type phishing, virus javascript ... sur des applications web 2.0, en espérant me faire comprendre le mieux possible !

### AJAX : Récapitulatif, les bases ...

AJAX est un regroupement de techniques et technologies qui permet de mettre en place des communications asynchrones, donc des communications avec un serveur sans rafraichir une page web, en gros.

## Web 2.0 Hacking : Introduction

Écrit par The\_NeTpSyChO

Dimanche, 03 Février 2008 12:53 - Mis à jour Mardi, 05 Février 2008 10:28

---

Dans le « web 2.0 » le langage considéré comme standard aux applications AJAX est le Javascript, je vous conseil donc de vous renseigner un minimum sur celui-ci avant de lire cet article. AJAX utilise un objet très spécifique de Javascript qui permet une communication avec un serveur : XMLHttpRequest. Les principales méthodes de cet objet son open(), send() et abort(). Il y a aussi d'autres objets requête que XMLHttpRequest, en effet Microsoft ne faisant rien aux normes utilise soit ActiveXObject("Msxml2.XMLHTTP") ou alors ActiveXObject("Microsoft.XMLHTTP"), mais bon cela ne change rien c'est un problème que l'on ne rencontre qu'une fois dans chaque application il suffit alors de tester le navigateur.

Je vous conseils aussi fortement de vous pencher sur le CSS, XML, DOM, ça risque de vous servir pour la suite !

Voilà maintenant que les généralités sont faites on peut commencer les choses sérieuses (tout doucement je vous rassure !).

### D'ou viennent les problèmes ?

Problèmes multiples : origines multiples. En effet les applications "web 2.0" souffrent, comme toutes les applications d'ailleurs, du fait que plus elles sont importantes plus le, ou les, développeur met de code plus il a de chance de faire des erreurs. Ensuite l'atout des sites web 2.0, pour l'utilisateur du moins, vient du faite qu'elle octroient une grande liberté (inclure des vidéos, des images, des liens etc ...). Hors plus l'utilisateur a de liberté plus il a de possibilité de faire des conneries, ou dans notre cas de détourner l'application.

Et tout est fait pour l'y aider, même le Javascript car en effet si vous vous êtes un peu renseigné sur ce langage vous devez savoir que c'est au fait un langage orienté prototype ( [allez voir sur wikipédia](#)

) ...

Petit exemple pratique :

On crée une méthode bidon :

```
XMLHttpRequest.methodBidon = function
```

```
{  
    return "coin";  
}
```

On clone XMLHttpRequest :

```
var pouet = new XMLHttpRequest();
```

Et voilà maintenant la méthode est disponible ici :  
pouet.methodBidon();

Je pense et j'espère que certains d'entre vous commencent à saisir par où je veux en venir, si ce n'est pas le cas ne vous inquiétez pas les explications arrivent très bientôt !

### Attaques basiques sur des applications AJAX : Prototype Hijacking

Nous allons maintenant étendre ce qui a été dit dans le chapitre précédent ... bienvenue dans le monde magique de la POP (programmation orientée prototype bien sûr), qui j'en suis sûr va beaucoup vous plaire !

Imaginons grâce aux méthodes de programmation OP que l'on s'amuse à faire ceci :

```
//On crée un clone de XMLHttpRequest
var xmlClone = XMLHttpRequest;
//On redéfinit XMLHttpRequest
XMLHttpRequest = function {
  //On appelle notre clone du XMLHttpRequest original
  this.xml = new xmlClone();
  return this;
}
```

Voilà ... là on est en plein dans le délire, on peut dire que si vous arrivez à injecter ceci c'est bon l'application est à vous ! Comment l'injecter ? XSS ça vous dit quelque chose. **La suite au prochain épisode ...**