Fiche Recap - Cross Site Scripting (XSS)

Introduction

Une attaque XSS est un type d'injection dans laquelle l'objectif est d'injecter des scripts malveillants directement à l'intérieur d'un site de confiance.

Dans le cas d'une application web, ça se traduit généralement par la capacité d'injecter un bout de javascript sur un site internet valide.

Il existe beaucoup de manière d'opérer ces injections et énormément de vecteurs de failles de ce type à l'intérieur d'un site web. Il s'agit d'un type d'attaque qui est donc très prisé pour un attaquant.

Le flow d'attaque consiste en l'envoi d'une URL à une cible qui contient une référence d'input utilisateur qui aura pour effet de lancer le script souhaité.

Le script étant techniquement généré par le site source qui est de confiance, le navigateur n'a aucun moyen de savoir qu'une attaque est en cours et donc il exécutera le script.

Etant donné que le script provient techniquement d'une source de confiance, le navigateur va donner accès au script à toutes les ressources du navigateur (cookie, tokens de session, ...) et autres informations sensibles qui peut être récupérée du côté du client.

L'objectif n'est pas forcément de voler une donnée, ça peut être de modifier le contenu de la page HTML, par exemple pour modifier le hash de vérification d'un fichier téléchargé, ou jouer des scénarios utilisateurs (cliquer sur des boutons, ...).

Types de XSS

De base, on considère deux types d'injections XSS, les injections côté client (reflétées) et les injections côté serveur (stockées). La différence réside dans la persistance de l'attaque.

Reflected XSS

Un XSS reflété apparait quand une entrée utilisateur est directement renvoyée par une application web sous la forme d'un message d'erreur ou d'un affichage direct sur la page de l'entrée en question ou d'une simple partie de celle-ci sans que l'entrée ne soit proprement affichée dans le navigateur et sans que l'entrée ne soit stockée de manière permanente sur le serveur.

Dans certains cas, la donnée ne va d'ailleurs même pas quitter le navigateur, ce qui peut rendre la détection de l'exploitation très complexe.

> Note : Il sera nécessaire de faire en sorte qu'un utilisateur clique sur notre URL forgée, il faudra donc rentrer dans une démarche d'ingénierie sociale.

Stored XSS

Un XSS stocké apparait quand l'entrée utilisateur est stockée sur le serveur distant, dans une base de donnée, un message sur un forum, des journaux, ce genre de choses...

L'exploitation réelle survient lorsque la victime accède à la ressource depuis son navigateur sans que celle-ci ne soit traitée au préalable.

> Note : Dans ce cas, pas besoin d'ingénierie sociale, chaque fois qu'un utilisateur va charger la page, le script sera exécuté.

Exemples :

Exemple 1 : Reflected XSS

```
<html>
                <head></head>
                <body>
                        <?php printf("Not found : %s", $_REQUEST["param"]); ?>
                </body>
</html>
```

Exemple 2 : Stored XSS

```
<html>
                <head></head>
                <body>
                                <?php
                                        if($_REQUEST["param"])
                                        {
                                                file_put_contents("./error_logs",
$_REQUEST["param"].PHP_EOL);
                                        }

                                        printf("%s",
file_get_contents("./error_logs"));
                                ?>
                </body>
</html>
```