Mayeul BOERI

# Documentation sur les attaques XSS (Cross-Site Scripting) et CSRF (Cross-Site Request Forgery)

### XSS (Cross-Site Scripting)

L'attaque XSS repose sur une faille qui permet à un attaquant d'injecter du code JavaScript malveillant dans une page web, code qui sera ensuite exécuté dans les navigateurs des utilisateurs qui visitent cette page. Cette attaque est possible en raison du manque d'isolation des données non fiables sur de nombreux sites.

Tout commence par l'injection du code malveillant. L'attaquant insère du code JavaScript dans un champ de formulaire, un lien ou un autre point d'entrée vulnérable sur le site. Ce code est soit stocké sur le serveur (stocké) ou exécuté directement sur la page (réfléchi) à travers une faille dans le code du site. Lorsqu'un utilisateur visite la page compromise, le code JavaScript malveillant s'exécute dans son navigateur, permettant ainsi à l'attaquant d'accéder aux cookies, de voler des informations sensibles ou de prendre le contrôle de la session de l'utilisateur.

Les navigateurs web exécutent automatiquement le code JavaScript inclus dans une page web, ce qui rend l'attaque XSS possible. Les cookies de session sont souvent stockés dans le navigateur et sont accessibles via JavaScript, fournissant ainsi un moyen pour l'attaquant de voler des informations sensibles.

Pour se protéger contre les attaques XSS, il est essentiel de valider et d'échapper correctement toutes les données utilisateur avant de les afficher sur une page. L'utilisation de l'en-tête Content Security Policy (CSP) permet de restreindre les sources de contenu autorisées sur une page web. Éviter l'utilisation de méthodes dangereuses comme `innerHTML` pour injecter du contenu utilisateur directement dans le DOM est également crucial.

### CSRF (Cross-Site Request Forgery)

L'attaque CSRF repose sur un mécanisme différent, exploitant la confiance accordée aux cookies de session. Elle vise à forcer un utilisateur à effectuer des actions indésirables sur un site web sans son consentement.

Tout commence par la préparation de l'attaque. L'attaquant crée une page web malveillante contenant des requêtes HTTP vers le site ciblé, telles qu'une demande de modification de mot de passe. Ensuite, l'attaquant incite l'utilisateur à visiter cette page, par exemple, en envoyant un lien via un e-mail. Si l'utilisateur a une session active sur le site ciblé, les requêtes malveillantes sont exécutées avec les privilèges de l'utilisateur, sans son consentement.

L'attaque CSRF tire parti du fait que les cookies de session sont automatiquement inclus dans chaque requête HTTP vers un site web. Lorsqu'un utilisateur est déjà authentifié sur le site, le navigateur envoie les cookies de session avec la demande, ce qui donne l'impression que la demande est légitime.

Pour se protéger contre les attaques CSRF, il est essentiel d'utiliser des jetons anti-CSRF (CSRF tokens) inclus dans les formulaires, empêchant ainsi les attaquants d'exploiter des demandes malveillantes. La vérification de l'en-tête Referer pour s'assurer que les requêtes proviennent de sources légitimes est une autre mesure importante. De plus, réservez l'utilisation de requêtes POST pour les actions sensibles, car elles sont généralement protégées contre les attaques CSRF.

### Exemples d'application

Les attaques XSS et CSRF ont un large éventail d'applications malveillantes. Dans le cas de XSS, un attaquant peut voler des cookies de session pour accéder au compte d'un utilisateur ou modifier le contenu d'une page web pour afficher des annonces ou des liens malveillants.

Quant à CSRF, un attaquant peut forcer un utilisateur à changer son mot de passe en visitant une page malveillante, ou encore faire en sorte qu'un utilisateur effectue un paiement sans son consentement en cliquant sur un lien. Ces attaques peuvent avoir des conséquences graves en matière de sécurité, d'où la nécessité de mettre en place des mesures de protection adéquates pour les prévenir.