**Comprendre le CSRF (Cross-Site Request Forgery)**

**1. Définition**

Le **CSRF (Cross-Site Request Forgery)** est une attaque qui force un utilisateur authentifié à exécuter une action non désirée sur un site web où il est connecté. Cette attaque exploite la confiance d'un site dans l'identité de l'utilisateur plutôt qu'une vulnérabilité spécifique du site lui-même.

**2. Fonctionnement**

Un attaquant profite du fait qu'un utilisateur est authentifié sur un site pour lui faire exécuter, à son insu, une requête malveillante. Le processus suit généralement ces étapes :

1. L'utilisateur se connecte à un site web légitime et obtient un cookie de session valide.
2. L'attaquant envoie à l'utilisateur un lien malveillant (email, site piégé, image contenant une requête cachée, etc.).
3. Si l'utilisateur clique sur ce lien ou visite la page piégée, son navigateur exécute une requête HTTP vers le site cible, en envoyant automatiquement les cookies d'authentification.
4. Si aucune protection CSRF n'est mise en place, le serveur traite la requête comme légitime et exécute l'action demandée.

**3. Exemple d'attaque CSRF**

**a) Site vulnérable**

Un site bancaire permet de transférer de l'argent avec une requête GET :

<a href="https://bank.com/transfer?amount=1000&to=attacker">Click here</a>

Si un utilisateur authentifié clique sur ce lien, son navigateur envoie la requête automatiquement avec ses cookies de session, transférant de l'argent sans confirmation supplémentaire.

**b) Variante avec formulaire caché**

Un attaquant peut également utiliser un formulaire automatique :

<form action="https://bank.com/transfer" method="POST">

<input type="hidden" name="amount" value="1000">

<input type="hidden" name="to" value="attacker">

<script>document.forms[0].submit();</script>

</form>

Si l'utilisateur visite la page contenant ce code, le formulaire est soumis automatiquement.

**4. Différences entre CSRF et XSS**

Le **CSRF** (Cross-Site Request Forgery) et le **XSS** (Cross-Site Scripting) sont deux attaques différentes bien qu'elles puissent parfois être confondues.

Le **CSRF** exploite la confiance qu'un site a envers un utilisateur authentifié. L'attaquant force l'utilisateur à exécuter une action non désirée sur un site où il est connecté, sans son consentement explicite. L'attaque repose sur le fait que les navigateurs envoient automatiquement les cookies d'authentification avec les requêtes.

Le **XSS**, en revanche, injecte du code malveillant (souvent JavaScript) dans une page web qui sera ensuite exécuté par le navigateur de la victime. Cette attaque vise directement l'utilisateur et permet à l'attaquant de voler des données sensibles, comme des identifiants ou des cookies de session.

En résumé, le **CSRF** abuse de la session de l'utilisateur pour exécuter une action malveillante sur un site de confiance, tandis que le **XSS** vise à injecter et exécuter du code malveillant dans le navigateur de l'utilisateur.

* **CSRF** exploite la relation de confiance entre l'utilisateur et le site.
* **XSS (Cross-Site Scripting)** insère du code malveillant dans une page légitime pour voler des données ou exécuter des actions.

**5. Méthodes de protection**

**a) Jetons CSRF**

Les sites sécurisés utilisent des **jetons CSRF** (CSRF tokens), générés aléatoirement et inclus dans les formulaires ou les requêtes sensibles.

<input type="hidden" name="csrf\_token" value="RANDOMLY\_GENERATED\_TOKEN">

Le serveur vérifie ensuite que le token soumis correspond bien à celui stocké pour l'utilisateur.

**b) Vérification du header "Origin" ou "Referer"**

Le serveur peut rejeter les requêtes provenant d'un site externe en vérifiant les en-têtes HTTP.

**c) Cookies SameSite**

L'attribut SameSite des cookies empêche leur envoi dans des requêtes cross-site.

Set-Cookie: sessionid=abc123; Secure; HttpOnly; SameSite=Strict

**6. Sources et ressources supplémentaires**

* OWASP : <https://owasp.org/www-community/attacks/csrf>
* Mozilla MDN : <https://developer.mozilla.org/en-US/docs/Glossary/CSRF>
* PortSwigger : <https://portswigger.net/web-security/csrf>