Attaque CSRF (*cross-site request forgery*)

Texte de Patrick Roberge – 29A

Définition Wikipédia :

« L'objet de cette attaque est de transmettre à un utilisateur authentifié une requête HTTP falsifiée qui pointe sur une action interne au site, afin qu'il l'exécute sans en avoir conscience et en utilisant ses propres droits. L'utilisateur devient donc complice d'une attaque sans même s'en rendre compte. L'attaque étant actionnée par l'utilisateur, un grand nombre de systèmes d'authentification sont contournés. »

Le but de l'attaquant est de changer l'état des données et non de soutirer de l'information. Voici un exemple concret : Vous êtes **connecté** à un forum quelconque (nous appelons cet onglet l'« onglet A »). Un autre onglet est ouvert, que nous appelons l'« onglet B ». Le script malveillant provient de l'onglet B et provoque une requête vers l'application de l'onglet A.

L'application de l'onglet A reçoit la requête et la traite normalement étant donné que la session est encore valide.

Comment est-ce possible?

Imaginons que Boris est un utilisateur régulier d'un forum et que Joe est un administrateur du même forum. **Boris arrive à connaître quelques liens permettant de faire des modifications** dans le forum. Il aimerait bien réussir un tour de force pour faire rigoler l'administrateur Joe. Boris lui envoie donc un message présentant une pseudo-image, qui contient un script. L'URL de l'image est un lien vers le script permettant d'effectuer une action pouvant être exécutée **uniquement** par un administrateur. Étant donné que Joe est un **administrateur connecté**, le script s'exécute normalement et Joe n'y voit que du feu!

Comment se protéger?

- Demander à l'utilisateur de répondre à une question avant d'exécuter la requête. On rencontre souvent ce type de processus. Par exemple, vous avez sûrement dû un jour reconnaître les devantures de magasins ☺. Cette action est une protection visant à éviter d'être pris au piège par un script malveillant.
- Valider le *Referrer information*.
- Vérifier la présence d'un jeton CSRF (*token*).
- Répondre à un test captcha.

En résumé, il importe de ne pas rester connecté en permanence sur une application web. Plusieurs moyens permettent d'éviter ce type d'attaque. Protégez vos applications et forcez vos sessions à se déconnecter après un temps d'inactivité.
