Titre de l'article : Nouvelle tendance : Les attaques CSRF

Auteur : Adrien Guinault, consultant sécurité XMCO
Courriel : adrien.guinault@xmco.fr

Date : Février 2007

Version complète en PDF : Actu Sécurité Février 2007

La relation d'authentification entre le client et le serveur au sein d'une application web constitue un enjeu majeur. La plupart des applications identifient correctement les utilisateurs et comprennent leurs requêtes.

Cependant, certaines ont encore beaucoup de mal à vérifier les origines des requêtes et les intentions réelles des clients. L'attaque que nous allons vous présenter est simple à mettre en place et pourtant, peu de webmasters y sont sensibilisés.

Qu'est ce que le CSRF??

Définition

CSRF ou plus communément appelée "**Cross Site Request Forgery**" est une classe d'attaques propre aux applications Web. Elle reste l'une des moins médiatiques malgré les effets dévastateurs qu'elle peut présenter. Issue d'un problème nommé "**Confused deputy**" [1] découvert par Norm Hardy en 1988, l'attaque relative aux applications Web fut officiellement révélée en Juin 2001 par Peter Watkins sur la liste de diffusion "Bugtrack". Elle fut surnommée "Cross Site Request Forgery" mais est plus communément appelé CSRF (prononcée *Sea Surf*) ou XSRF.

Cette technique exploite la confiance qu'une application montre à l'égard de ses clients. Le but est de forcer le navigateur de la victime à envoyer une requête silencieuse à l'insu d'un internaute.

Au premier abord, cette attaque peut sembler difficile à mettre en place. Détrompez-vous! Elle peut être menée par n'importe qui et d'une manière extrêmement simple. Pour cela il suffit d'insérer un script dans une page web ou de le camoufler dans un e-mail. En suivant le lien contrefait, le navigateur de la victime va exécuter une requête vers un site sur lequel la victime est authentifiée (voir démonstration plus bas).

Méconnue par la plupart des RSSI, elle vise particulièrement les sites web dont les structures des requêtes utilisées sont prédictibles.

Sachez, avant de vous plonger dans des explications détaillées et des exemples concrets, que le taux de réussite de ce genre d'attaque est bien plus élevé que ce que l'on imagine...

XSS et CSRF?

Le nom ressemble étrangement à la technique "**XSS**" (Cross Site Scripting). Il est fréquent que ces deux attaques soient assimilées alors qu'elles sont diamétralement opposées.

Le XSS est une attaque construite dans le temps, menée étape par étape (vol de cookie puis réutilisation), qui a pour but d'injecter du code dans un document HTML afin d'abuser le navigateur client. L'attaque cible un site précis qui possède un problème de validation de certaines entrées utilisateur.

Le CSRF est une attaque instantanée. Elle ne repose pas sur l'exécution d'un script dans un navigateur. Son but est d'exécuter une action non désirée par le client sur un site où la victime possède un accès privilégié.

Concrètement, le XSS est réalisé pour voler un cookie de session alors que le CSRF va utiliser le cookie de session (sans que le pirate ne connaisse sa valeur) et la confiance établie entre le site web et le client afin d'exécuter une requête légitime mais toujours à l'insu de l'utilisateur abusé.

La portée de cette attaque est donc plus étendue. Un site peut se protéger du XSS en utilisant un filtre sur les entrées utilisateur alors que la requête exécutée avec le CSRF paraîtra totalement légîtime. Nous sommes au centre du problème...

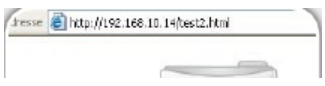
Le CSRF se base sur la confiance attribuée aux utilisateurs par l'application.

Comment ça marche?

Explications de l'attaque avec une balise

Les **balises IMG** représentent l'un des vecteurs d'attaques les plus répandus.
Afin de bien comprendre la portée de cette faille, prenons un exemple simple. Le code suivant est une page web html qui affiche à la suite du mot "XMCO PARTNERS", une image provenant de notre site web.

```
<html>
<p>XMCO PARTNERS :
<img
src="http://xmcopartners.com/images/access.gif
" /></p>
</html>
```



Version complète en PDF

Autres articles

- Les vers XSS
- Les attaques Cross Site Scripting
- Tous les articles XMCO
- *S'abonner à la newsletter*

Le cabinet XMCO

- La société
- Contacts

Web2PDF
converted by Web2PDFConvert.com

Le navigateur qui affiche cette page va aller télécharger l'image "access.gif" sur le serveur qui héberge le site "xmco.fr" sans que l'utilisateur ne s'en aperçoive. L'opération est transparente pour l'utilisateur...

La même requête avec un nom d'image erroné donne alors le résultat suivant :

Dans ce cas, la requête " GET /images/acceszzzzz.gif" n'a pas aboutie mais a bien été interprêtée par le serveur qui a renvoyé une erreur " 404 not found".

Le navigateur ne fait aucune différence entre une requête GET d'une page web ou d'une image. On voit ici quel impact pourrait avoir une requête particulièrement travaillée et envoyée à l'insu de la victime. En injectant le code suivant, on peut imaginer les conséquences si l'utilisateur est préalablement loggué sur le site en question. On force ici l'utilisateur à acheter 100 télévisions...

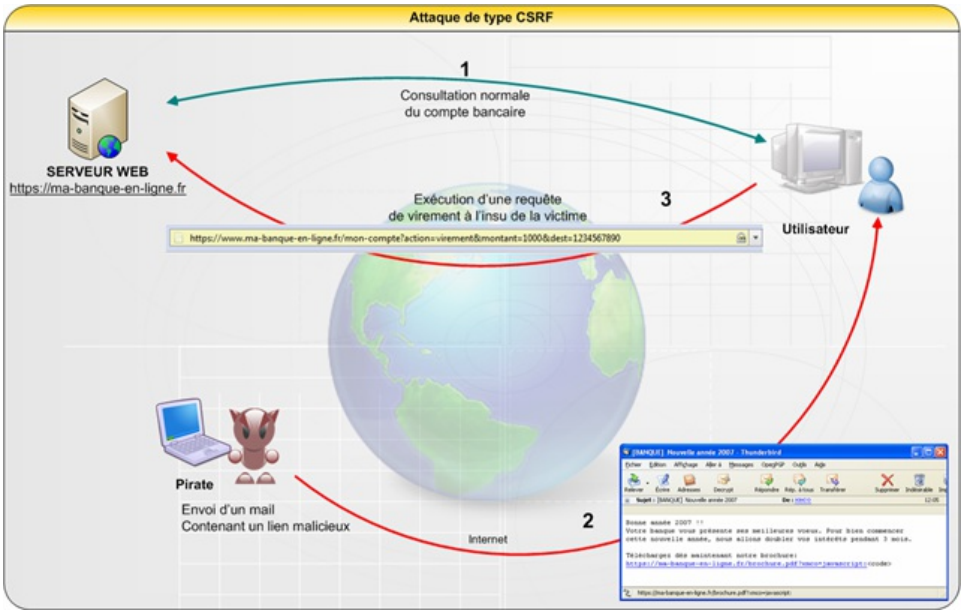
```
<img src= http://www.achat-en-ligne.com/index?buy=tv&nb=100&confirm=1>
```

Un scénario d'attaque

Le scénario d'une telle attaque est relativement proche d'une tentative de vol de session par une attaque XSS. Tout l'intêret de cette technique va être d'inciter la future victime à visiter une image factice chargée d'envoyer une requête HTTP au serveur.

Imaginons le cas suivant : vous êtes connecté sur le site de votre banque afin de vérifier le solde de votre compte en banque. Un ami qui a pris soin d'étudié le type de requête envoyé lors d'un transfert d'argent entre deux comptes, vous envoie un message par email pour vous inciter à visiter une adresse malicieuse.

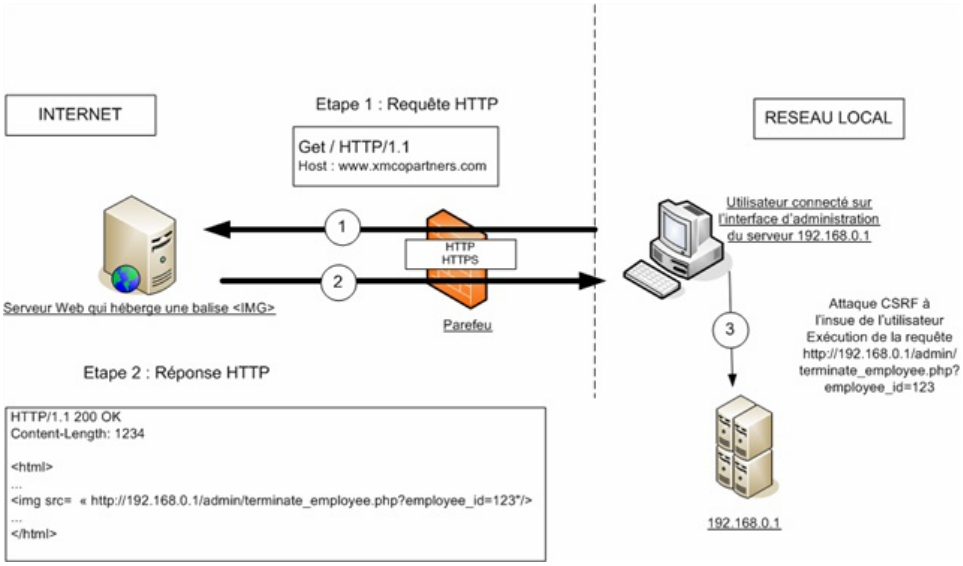
Comme la plupart des internautes crédules, un email correctement rédigé avec un sujet important ou intéressant va générer un vif attrait pour le lien envoyé.
Une fois l'adresse suivie, une image s'affiche dans votre navigateur. Une "iframe" cachée soumet un formulaire au site de votre banque (sur lequel vous êtes identifié) sans que vous vous en rendiez compte.



L'attaque sera particulièrement intéressante à mener sur un réseau local. Les utilisateurs se connectent une seule fois sur les différentes applications de l'intranet et y restent connectées toute la journée tant que le navigateur n'est pas fermé. De même, les points d'accès et les applications HTTP sensibles (les administrateurs y sont le plus souvent connectés en permanence) peuvent être ciblés.

Le schéma suivant montre la portée de cette attaque. En connaissant particulièrement le réseau local d'une entreprise (exemple : un ancien administrateur licencié), un pirate externe pourrait exécuter une requête à l'intérieur d'un réseau. Cette technique a été également présentée sous le nom de "**Drive-by-pharming**" lorsque l'attaque cible les routeurs (Box) des particuliers.

Web2PDF
converted by Web2PDFConvert.com



Cette attaque requiert certaines conditions que nous présentons dans la suite de cet article.

Mon application est-elle vulnérable?

Conditions requises

Plusieurs conditions doivent être réunies pour permettre une éventuelle attaque. Les sites qui implémentent des requêtes "POST" sont moins exposés. En effet, les requêtes "GET" sont facilement prédictibles. Les informations transitent dans l'URL, il suffit donc de forger une URL afin d'envoyer une requête valable au serveur web.

Par ailleurs, le pirate doit être certain que la victime est authentifiée sur le site et qu'elle possède un cookie. Le site web cible ne doit pas implémenter une seconde étape lors de la validation des actions et les paramètres envoyés doivent rester statiques.
Si plusieurs pages de formulaires sont utilisées, le pirate devra construire un script bien plus évolué qu'une simple URL. L'application sera donc moins exposée.

Les conséquences d'une attaque

La plupart des services fournis par une application web peuvent ainsi être exploité par cette attaque : changement de mot de passe, post de message sur un forum, achat en ligne, achat d'action sur un site boursier, envoi d'une carte de voeux virtuelle, envoi d' e-mails... Ce genre d'attaques est souvent perpétré à l'encontre des forums. Il utilise des balises images pour dissimuler le code.

On peut maintenant imaginer les conséquences d'une attaque plus évoluée. Une victime, connectée à son site en ligne visite un site pirate qui envoie une requête HTTP destinée à créditer son compte sans que cette dernière n'ait spécifiquement choisi d'effectuer cette action. Le site de la banque voit seulement un utilisateur connecté qui réalise une opération. L'application autorise la requête car elle accorde toute sa confiance à l'utilisateur (problème de non répudiation).

Un pirate pourra également forcer la victime à poster un ver sur un forum, ou à relayer une requête lourde vers un site choisi pour causer un déni de service. Pire encore, si le pirate utilise cette attaque pour passer un ordre boursier à l'insu d'un utilisateur, comment ce dernier pourra-t-il prouver qu'il n'est pas l'initiateur de la requête ?

Les solutions

Le Referer

Certains amateurs du protocole HTTP ont sans doute immédiatement pensé au " referer ", option qui identifie l'origine de la requête. Cette fonction, justement utilisée pour parer à ce genre de problème, n'est malheureusement pas mise en place par tous les internautes.
Cette solution n'est donc pas fiable. Elle fonctionnera dans certains cas. En revanche, le serveur sera obliger de refuser chaque requête qui ne possède pas ce champs. Par ailleurs, les pirates peuvent également bloquer l'envoi du " Referer ".

Utilisation d'un secret

La seule méthode vraiment efficace consiste à utiliser des tokens aléatoires (secret) sur toutes les pages sensibles. Ce jeton doit être envoyé au serveur (en champs caché) lors de la soumission d'un formulaire ou d'actions critiques. Si l'URL envoyée ne contient pas ce nombre aléatoire qui ne peut être prédit par l'attaquant, le serveur web ne traitera pas cette dernière.

Des noms de variables aléatoires

Une autre possibilité réside dans le fait que le serveur web implémente une table de nombre aléatoires qui sert à définir le nom d'une variable en fonction d'une session donnée. Dans ce cas, l' information ne peut pas être prédit par le pirate.

Double validation des actions

Une dernière possibilité consiste à obliger l'utilisateur à valider chaque action critique par la soumission de son mot de

Web2PDF
converted by Web2PDFConvert.com

passe. Une fois que la requête d'ordre de transfert d'argent a été effectuée, l'utilisateur doit confirmer avec un paramêtre (dont le pirate ne dispose pas). L'attaque est alors parée.

Les solutions de contournements du coté du client

Toute la problématique réside dans le fait d'empêcher le navigateur d'effectuer des requêtes sans l'autorisation préalable du client. Voici quelques conseils précieux pour vous aider à parer ce genre d'attaque:
-ne pas utiliser un client mail qui interprête les codes HTML.
-ne pas sauvegarder les identifiants dans le navigateur.
-ne pas utiliser la fonction " remember me " proposée par de nombreux sites.
-ne pas suivre les liens suspects.
-se déconnecter lorsque vous avez fini de visiter les sites sensibles.

Webographie

[1] Article complet dans l'Actu-Secu n°11 (Février 2007).
http://www.xmco.fr/actu-secu/actu_secu_fevrier2007.pdf

[2] Explications du problême "Confused Deputy"
http://www.cis.upenn.edu/~KeyKOS/ConfusedDeputy.html

[3] Le white paper de la société Isecpartners
http://www.isecpartners.com/documents/XSRF_Paper.pdf

[4] FAQ du groupe cgisecurity :
http://www.cgisecurity.com/articles/csrf-faq.shtml

Web2PDF
converted by Web2PDFConvert.com