

Date: Samedi 17 novembre 2007 &agrave; 17:16:26

Sujet: 3 Sécurité et Hacking

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

Les injections SQL peuvent servir &agrave; se connecter sur le compte d'une autre personne sans son mot de passe o&ugrave; encore &agrave; trouver des mots de passe.

C'est un type de faille tr&egrave;s r&eacute;pandues sur le web et dont le fonctionnement est assez simple.

Il est indispensable de comprendre les injections SQL si on veut correctement prot&eacute;ger un site web.

Cet article s'adresse &agrave; tout le monde, que vous vouliez en apprendre plus sur la s&eacute;curit&eacute; informatique ou que vous souhaitez s&eacute;curiser votre site web.

Les injections SQL Sommaire :

[1 - Description de la faille](#)

[2 - Se loguer sans mot de passe avec SELECT](#)

[3 - Se loguer sur le compte d'une autre personne](#)

[4 - Trouver le mot de passe avec LIKE](#)

[5 - Corriger la faille](#)

1 - Description de la faille On peut faire beaucoup d'injections SQL diff&eacute;rentes, 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 mor&ccedil;eaux de codes SQL dont les informations sont forc&eacute;ment vraies (par exemple : 1=1).

Code HTML du formulaire de connection : `&lt;form method=&quot;post&gt;&lt;input type=&quot;text&quot; name=&quot;pseudo&quot; /&gt;&lt;input type=&quot;password&quot; name=&quot;password&quot; /&gt;&lt;input type=&quot;submit&quot; name=&quot;Connexion&quot; /&gt;&lt;/form&gt;`

La plupart des sites utilisent un formulaire de connection similaire a celui-ci. Les informations envoy&eacute;es dans ce formulaire sont ensuite trait&eacute;es par une page en PHP qui v&eacute;rifie les informations entr&eacute;es avec celle pr&eacute;sent&eacute;es dans la base de donn&eacute;es.

2 - Se loguer sans mot de passe avec SELECT Les variables que vous avez ente&eacute;s dans le forumaire vont etre ensuite utilis&eacute;es 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 antislashes ( `\` ) devant les quotes ( `'` ) et les doubles quotes ( `&quot;` ) pour &eacute;viter les injections SQL.

Mais cette fonction n'est pas toujours activée; par les h&eacute;bergeurs ce qui engendre une vuln&eacute;rabilit&eacute;.

Dans ce premier exemple les logins et mots de passe sont stock&eacute;s dans une table users de la base de donn&eacute;e et nous allons prendre comme login it's me et comme password ah'ah"ah Si get\_magic\_quotes\_gpc() est activ&eacute;e, la requete sera interpr&eacute;t&eacute;e comme cela :

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

Si get\_magic\_quotes\_gpc() est d&eacute;activ&eacute;e, la requete sera interpr&eacute;t&eacute;e 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 donn&eacute;e et que le mot de passe entr&eacute; correspond au mot de passe de la base de donn&eacute;e.

Pour l'instant vous ne voyez pas le danger que provoque la d&eacute;activation de get\_magic\_quotes\_gpc(), mais vous allez comprendre.

Nous allons maintenant injecter des donn&eacute;es 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 caract&egrave;res doivent etre entour&eacute;s de quotes ' dans les requetes SQL)
- 'z'='z'
- 42&lt;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 tromp&eacute; avec les quotes, on va bien entrer 'OR 'a'='a'

Vous pensez peut etre qu'il y a un quote ' en trop au d&eacute;but 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 d&eacute;but et que je n'en ai pas mis a la fin.  
En effet, il y a d&eacute;ja 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 forc&eacute;ment vrai.

Quand il n'y a rien entre 2 quotes "", cela veut dire que la chaine de caract&eacute;re est vide mais cela ne nous sera jamais utile, a part si vous tombez sur un site web qui ne v&eacute;rifie pas que les champs soient remplis et qu'un mec n'a rien mis en mot de passe. Autant vous dire que les chances sont tr&eacute;s 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 &eacute;gale &agrave; 'a'le pseudo sera a chaque fois selectionn&eacute; et vous serez connect&eacute; 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 donn&eacute;e 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 probl&eacute;mes avec les quotes ' mais on devra tout de m&eacute;me injecter une information de type caract&eacute;re entour&eacute;e 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 ferm&eacute;s au bon endroit.  
Le quote a la fin de login="OR 1=1'est invalide car il ouvre un quote qui est mal plac&eacute;.  
Maintenant vous comprenez pourquoi j'ai inject&eacute; 'a'='a' tout a l'heure.

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

Le problème c'est qu'on a entré 'OR 'a'='a en pseudo, et que le site nous a 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 même beaucoup plus intéressant.

Dans notre exemple, on va essayer de se connecter au compte de TomSawyer alors qu'on ne connaît 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 récemment 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 être 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 à galemment; la requête ci-dessous comme le quote ' après le # n'est pas interprétée;.

Code : SQL

```
SELECT login FROM users WHERE login='TomSawyer' AND password='OR 1=1'
```

La requête sélectionne le pseudo TomSawyer si le mot de passe est vide ( " ) ou si 1=1, comme on l'a déjà vu comme 1 est toujours égal à 1, le pseudo TomSawyer sera sélectionné; et nous nous connecterons à son compte.

La vérification du mot de passe ne sert plus à rien dans cet exemple comme on arrive à 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 sélectionné; 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 v&eacute;rifi&eacute; (&agrave; cause du quote '). Mais mettez quelque chose car la plupart des scripts PHP v&eacute;rifient que les champs du formulaire soient bien remplis.

La requete SQL sera donc interpret&eacute;e comme celle qui suit (on a simplement supprim&eacute; tout ce qui se trouvait apr&egrave;s 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 v&eacute;rification. C'&eacute;tait 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 d&eacute;mand&eacute;.

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

La premi&egrave;re chose a faire quand on cherche un mot de passe, c'est de trouver sa longueur gr&acirc;ce &agrave; 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 :

&nbsp; Code : SQL

SELECT login FROM users WHERE login='TomSawyer' AND LENGTH(password)=4#

AND password='blablabla' qui &eacute;quivaut a la requete suivante apre&egrave;s suppression des instructions apr&egrave;s 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 caract&eacute;res.

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 crypt&eacute; en MD5, sa longueur sera 32 et il sera tr&egrave;s difficile &agrave; trouver.

Maintenant qu'on connait la longueur du mot de passe, on va essayer de le trouver.

Je vous pr&eacute;viens tout de suite, c'est pas tr&egrave;s marrant &agrave; faire et c'est tr&egrave;s r&eacute;p&eacute;titif. Vous pouvez

d'ailleurs créer un script qui le fait à 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 être :

Code : SQL

```
SELECT login FROM users WHERE login='TomSawyer' AND password  
LIKE 'a%#' AND password='blablabla'
```

équivalent :

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 à 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'alphabet).

Si vous ne trouvez toujours pas essayer des chiffres (1,2,3). Attention à 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 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 deuxième (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 deuxième lettre n'est pas a.

On va donc essayer avec b comme deuxième 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'à ce que cela marche, quand vous avez essayé les lettres de a à z, essayez les

chiffres de 0 à 9.

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

Les dernières lettres du mot de passe sont souvent plus faciles à 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 (4ème), on tentera directement avec un o plutôt que d'essayer encore une fois toutes les lettres de l'alphabet en commençant par a.

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

Comme vous avez pu le voir la méthode est longue, mais c'est plus rapide quand on connaît la longueur du mot de passe et comme je vous l'ai dit, c'est assez simple de concevoir un script qui teste tout à votre place

Le script sera bien plus rapide que les logiciels qui testent 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 êtes webmaster et que votre hébergeur a désactivé la fonction `get_magic_quotes_gpc()` alors vous pouvez être victime d'injection SQL.

Pour vous protéger vous et vos membres, utilisez la fonction `addslashes()` qui rajoute des anti-slashes devant les quotes ' si `get_magic_quotes_gpc()` est désactivé.

Par exemple :

#### Code : PHP

```
$login = addslashes($_POST['login']);  
$password = addslashes($_POST['password']);
```

Si vous êtes un utilisateur, je vous conseille de bien choisir vos mots 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>