SQL Injections

Was sind SQL Injections?

SQL Injections sind eine Methode, mit welcher man Sicherheitslücken im PHP Code und von SQL ausnutzt. Ein schlecht programmiertes PHP Formular gibt die Eingaben direkt und nicht validiert an ein SQL Statement weiter. Dieser SQL String wird dann ausgeführt. Wenn nun die Eingabe aus dem Formular SQL Code enthält, dann wird dieser automatisch und ohne Überprüfung mitausgeführt. Dies kann zu falschen oder gelöschten Daten führen. Es ist ebenfalls ein Verstoss gegen die Datensicherheit, da sich so Hacker Zugang zu Login geschützten Bereichen verschaffen können.

Wie geht man vor, um SQL Injections durchzuführen?

Zuerst muss man die Architektur der Datenbank kennen um diese zu Manipulieren. Man muss zum Beispiel wissen wie die Tabellen heissen, wie viele Spalten es gibt und wie die Spalten benennt sind. Diese Informationen kriegt man durch erzwingen von Fehlermeldungen mit nicht validen Eingaben (String bei INT Feld usw.) oder GET Parameter, welche mitgegeben werden. Es gibt jedoch auch Tools, welche diesen Schritt übernehmen und die benötigten Informationen liefern. Hat man einmal diese Informationen, muss man in einem Eingabe Feld oder GET Parameter nur die richtige SQL Syntax eingeben und schon kann man Daten ausgeben, manipulieren oder sogar löschen.

Wie kann ich mein Programm schützen?

Es gibt verschiedene Wege, sich vor SQL Injections zu schützen. Anfänglich sollte man nur Eingaben zulassen, welche auch möglich sind, d.h. bei einem Feld für eine Postleitzahl sollen erstens nur Zahlen eingegeben werden können und nur eine gewisse Länge in diesem Falle vier. Dadurch kann kein brauchbarer SQL Code eingegeben werden wodurch das SQL Statement manipuliert werden kann. Bei Email Feldern kann man den User dazu zwingen eine korrekte Email Adresse einzugeben. Das Problem bei Textfeldern ist, alle Zeichen müssen zugelassen sein. Jedoch kann man die Länge Beschränken. Ein Namensfeld muss nicht länger als 15 Zeichen sein, dadurch wird die Gefahr minimiert. Es gibt bei PHP ebenfalls die Möglichkeit SQL zu escapen mit Funktionen wie zum Beispiel real_escape_string.
Die wohl schönste und am meisten verbreitete Methode ist jedoch Prepared Statements.

Prepared Statements

Prepared Statements sind wie der Name schon verrät vorbereitete Abfragen. Das heisst, der SQL Befehl wird soweit vorbereitet, dass nur noch die Parameter mitgegeben werden. Die Datenbank überprüft dann, ob das SQL Statement verändert wurde und escaped allfällige SQL Eingaben. Dadurch kann das SQL Statement nicht mehr manipuliert oder erweitert werden.

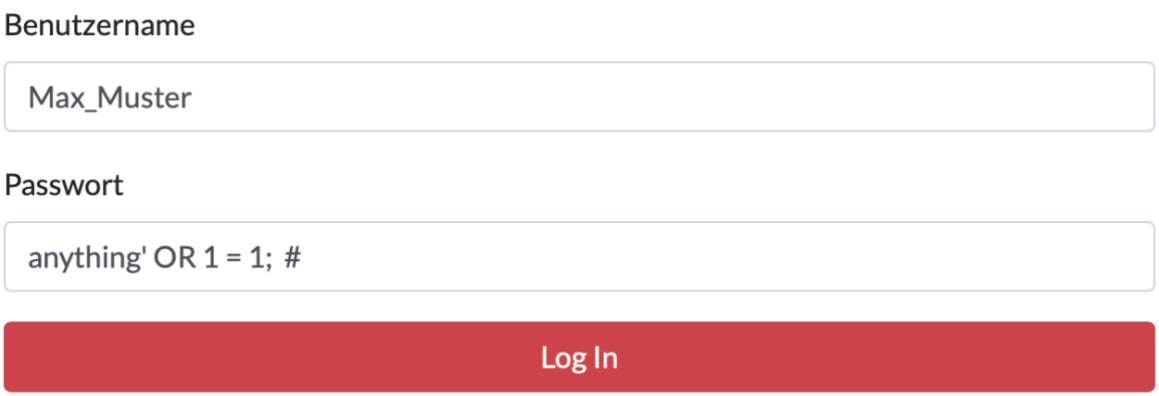
Beispiele

Angreifbarer Code

```
$sql = $dbcon->prepare("Select * from users where name = '$user' AND pw = '$pass'");
$sql->execute();
$data = $sql->fetchAll();
```

Hier sieht man, dass die Werte aus den Eingabefeldern ($ Variablen) direkt in den SQL String eingefügt werden, und dieser String dann ohne validieren ausgeführt wird. Dadurch kann der SQL String beliebig angepasst werden.

Angriff



Durch diese Eingabe wird das aktuelle Statement:
`Select * from users where username = '$user' AND password = '$pass'`
Zu folgendem:
`Select * from users where username = 'Janosch' AND password = 'anything' OR 1 = 1; #'`

Nun wird der User Janosch zurückgegeben egal ob das Passwort stimmt oder nicht. Dadurch kann sich der Hacker einloggen.

Sicherer Code

```
$sql = $dbcon->prepare("SELECT name, pwd from users where name = ? AND pwd = ?");
$sql->execute(array($user, $pass));
$data = $sql->fetchAll();
```

Nun weiss SQL, dass nur ein SELECT mit einem where und einem AND ausgeführt wird und dass kein OR danach kommt dadurch wird nur bei einem korrekten Passwort der Benutzer zurückgegeben.