16.3 Sicherheit SQL

Ein dramatisches Problem passiert immer wieder im Zusammenhang mit Datenbanken: Und zwar das Einschleusen von schädlichen SQL Code (SQL Injection).

Beispiel für eine ungefilterte SQL Abfrage

Ein Firmenname wird mittels POST an das Script übergeben. Der SQL-Code ist eine einfache INSERT Anweisung. Im Anschluss wird. Der SQL Code wird mit **`mysqli_query()`** *an die Datenbank geschickt.*

```
$firma = $_POST["Firma"];
$sql = "INSERT INTO FirmenDB (Firma) VALUES ('$firma')";
$ergebnis = mysqli_query($verbindung, $sql);
```

Wenn ein Benutzer nun

`Rosi's Imbiss`

eingibt, dann wird wegen dem Apostroph ein Fehler bei der SQL Abfrage entstehen. Das ist noch nicht so schlimm, problematischer wird es, wenn ein Hacker eine SQL Injection einschleust, z. B.:

`'); DELETE FROM FirmenDB --`

Dann wird folgender SQL Befehl abgeschickt (der die gesamte Datenbank löscht).

```
INSERT INTO FirmenDB (Firma)
VALUES ('); DELETE FROM FirmenDB --')
```

*Um dem also entgegen zu wirken, muss man unbedingt den Apostroph escapen. Zusätzlich sollten auch die kritischen SQL Zeichen - _ * % gewandelt werden. Entweder man nutzt dafür str_replace() oder die Funktion mysqli_real_escape_string() für MySQL.*

```
$firma = str_replace("'", "", $_POST["Firma"]);
$firma = mysqli_real_escape_string($_POST["Firma"]);
```

Weitere Sicherheitsmaßnahmen (für MySQL und phpMyAdmin):

***Rollenbasierte Berechtigungen**: Je nach Zweck sollte man eigene Benutzer mit dem niedrigsten Level an Rechten anlegen. z. B. für eine einfache Suchabfrage reicht ein Benutzer der nur die Rechte SELECT für die Datenbank (noch besser, nur für die Tabelle hat). Root niemals im PHP-Code verwenden!*

***Starke Passwörter**: mit mind. 8 Zeichen, Zahlen und Zeichen gemischt, keine Wörter die auch im Wörterbuch stehen usw. - hoch komplexes Passwort für root.*

***Datenbank-Administration**: Verschlüsselungen (z. B. OpenSSL, FIPS-Modus) aktivieren, Server-Sicherheit berücksichtigen, Richtigen Zeichensatz wählen (z. B. UTF-8), Plug-Ins und Erweiterungen auf das notwendigste reduzieren.*

***PHP-Sicherheitseinstellungen in der php.ini:** Funktionen deaktivieren, Dateizugriff auf externe URL unterbinden, Safe_mode einschalten, Zugriffsteuerung*