SQL-Injection

DB-Implementierungen

Luca Krawczyk 11.02.2025

```php
// Get the username and password from the POST parameters
$user = $_POST['user'];
$pwd = $_POST['pwd'];
$pwd_hash = hash( algo: 'sha256', $pwd);



$query = "SELECT * FROM login WHERE login.user = '".$user."' AND login.pwd = '".$pwd_hash."'";
$exec = $pdo->query($query);
$user = $exec->fetch();
```

```php
// Get the username and password from the POST parameters
$user = $_POST['user'];
$pwd = $_POST['pwd'];

// Prepare a SQL statement to select the user from the database
$statement = $pdo->prepare( query: "SELECT * FROM login WHERE user = :user");
// Execute the SQL statement with the username parameter
$result = $statement->execute(array('user' => $user));
// Fetch the user from the result set
$user = $statement->fetch();

// Check if the user exists and the password is correct
if ($user !== false && password_verify($pwd, $user['pwd'])) {
```

Prepared Statements

- Vorlage für SQL-Abfragen
- Vorab an DB-Server übermittelt und verarbeitet
- Ausführung Prepared Statement setzt Werte in die Vorlage ein

Quellen

- Vorlesung “IT-Sicherheit” - Kevin Drieschner WS 2024/25
- PHP MySQL Prepared Statements
  - https://www.w3schools.com/php/php_mysql_prepared_statements.asp
  - Abgerufen 09.02.2025

GitHub

github.com/luca910/SQL-Injection-SS25