ctfshow吃瓜杯 misc writeup

原创

AshMOB 于 2021-11-15 20:55:53 发布 218 收藏 1

分类专栏： ctf比赛wp 文章标签： 安全

本文链接：https://blog.csdn.net/ashMOB/article/details/121343260

ctfshow吃瓜杯 misc writeup

基本还是看着大佬wp去复现 ctfshow吃瓜杯 八月群赛 WriteUp/WP_是Mumuzi的博客-CSDN博客

Misc游戏签到

游戏题，比较看脸，因为遇到太多次靶场死机所以直接抄flag上去了

She Never Owns a Window

下载下来打开发现是一个文本文件，但是有许多空白换行和tab

提示说不是SNOW（**Steganography** -SNOW- AVariation:这是一种创新的 隐写 技术，可用于隐藏两个单词之间的空格后面的文本数据。 它是流行的 隐写 术 工具 的变体 - SNOW [Steganographic Nature Of Whitespace] 由 Matthew Kwan 开发，他曾经使用“空格”和“制表符”键将 ASCII 数据隐藏在尾随空格后面。）

mumuzi的wp说是Whitespace这个东西，wiki后得知是一种利用空格换行tab来进行编程的语言Whitespace - 维基百科，自由的百科全书 (wikipedia.org)

通过这个网址可以运行这段文字的程序Whitelips the Whitespace IDE (vii5ard.github.io)

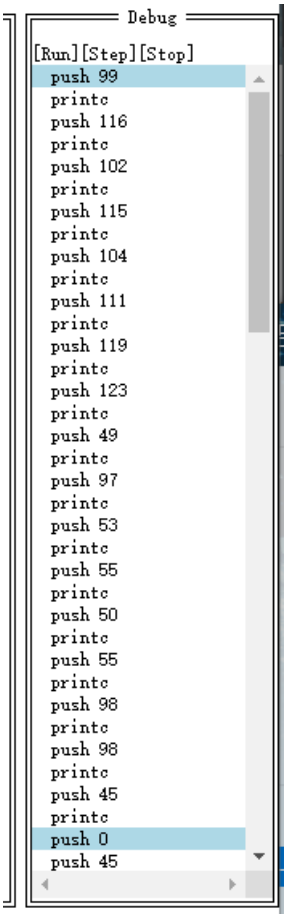
whitespace这个编程语言主要是利用栈来执行，一些命令如下

```
push ;数据压栈
dup ; 复制栈顶的数据并压入栈中
add ;弹出栈顶的两个数据相加后压回栈中
printc ;将栈顶元素弹出并输出
drop ; 将栈顶元素弹出栈
end ; 结束
```

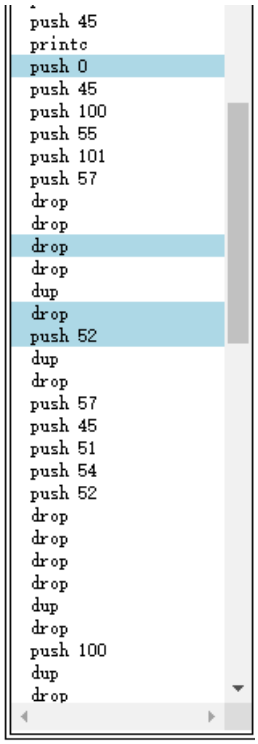
如图

注意到flag缺失。

不难看出该程序前期都在输出字符

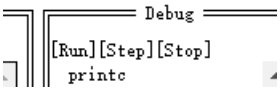


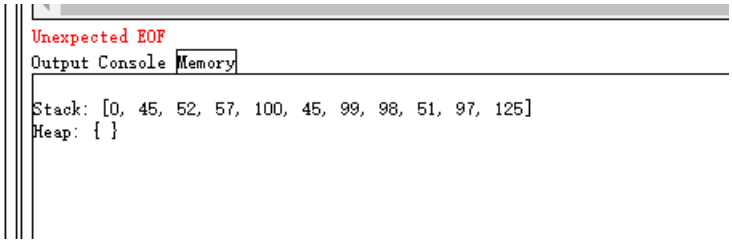
之后的某个段就只drop而不是printc了，剩余的flag就在这



因而我们要做的就是将drop换为printc，因为没学过whitespace就直接记下每次drop出去的数字然后放py里chr

栈中的数据可以在这里看到，按step在每次drop时观察栈的变化，记录





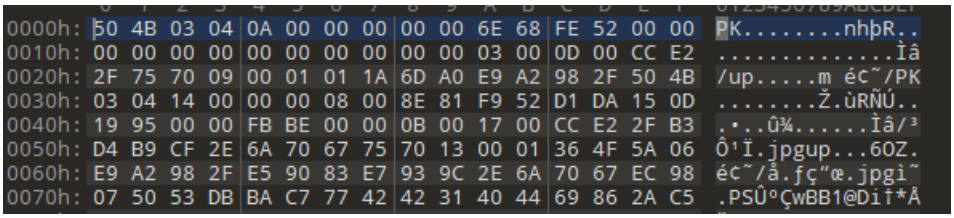
jio本：

```
#注意是动态flag
te=[57,101,55,100,45,52,52,54,51,45,57,100,56,48,45,50,53,48,99,99,102,53,97,98,102,51,97,125]
flag=""
for i in te:
    flag+=chr(i)

print(flag)
```

吃瓜

下载的解压出jpg，但是实际是一个压缩包



改后缀解压，得到一个图片和一个文本文件，文本文件data:image啥的具体看浅析data:image/png;base64的应用 - Angel_Kitty - 博客园 (cnblogs.com)

总之复制后浏览器直接能看到一个二维码

...

内容|
cfhwc19abika_etso{h_u_e_ui1}

解码得

猜测是栅栏密码，验证后在栏数为14时得到正确flag

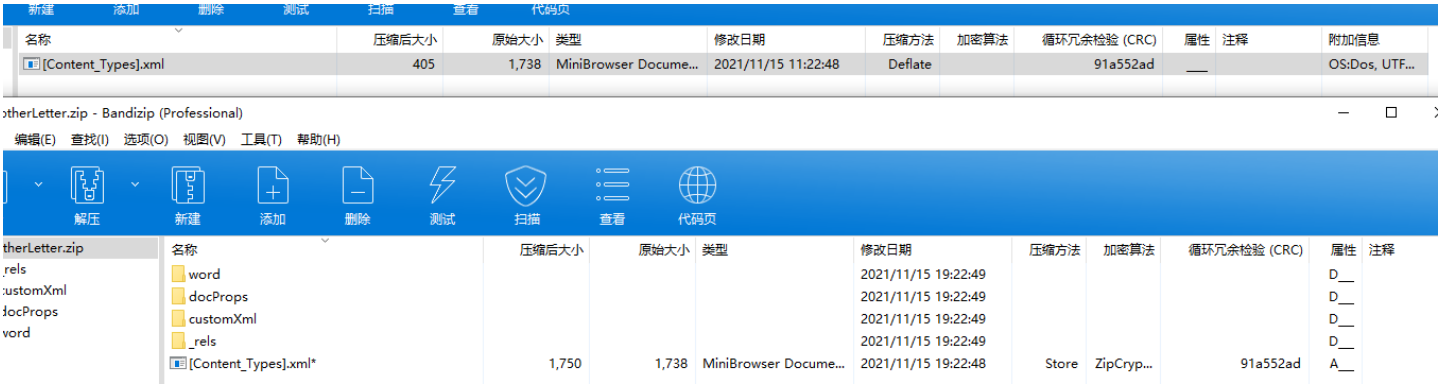
Dinner of Cyanogen

下载下来解压后是俩doc

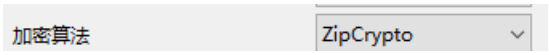
Anotherletter那个打不开，letter打开后哔哩吧啦讲了一堆要跑路的事，还留了个flag的头和之前的电话号码

俩都改成zip，doc本质就是压缩包

发现这俩压缩包里有不少文件的crc一样，尝试明文爆破

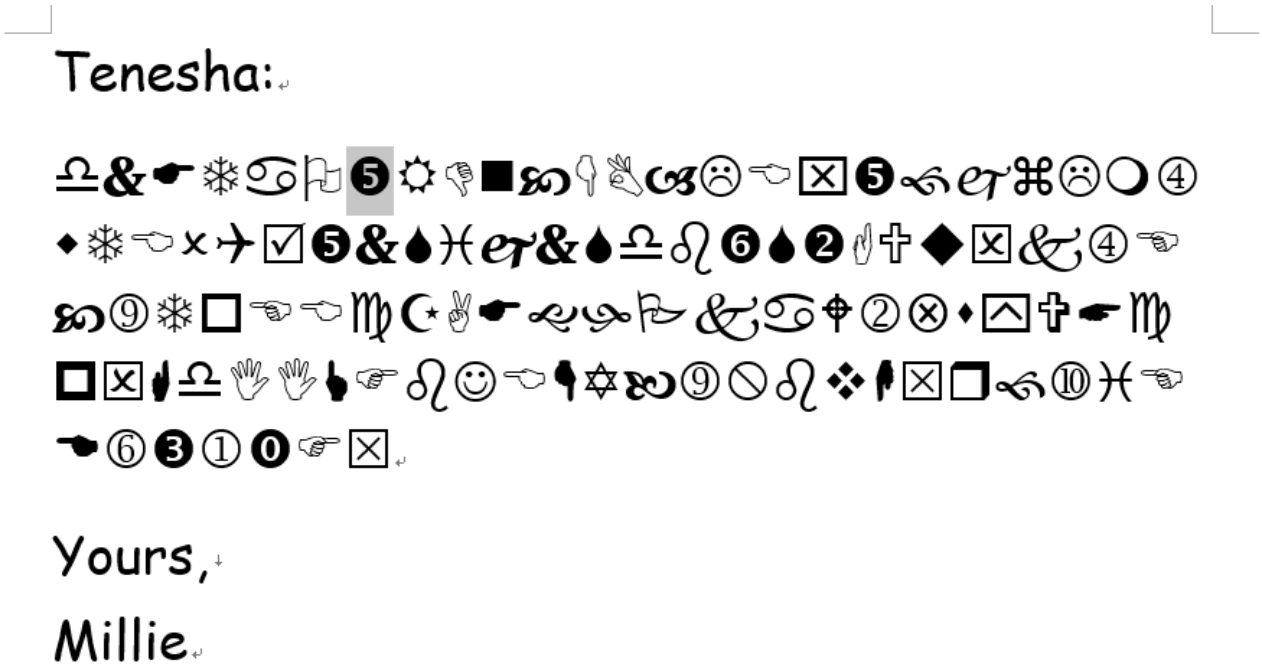


这俩文件压缩方法不同，要重新压缩

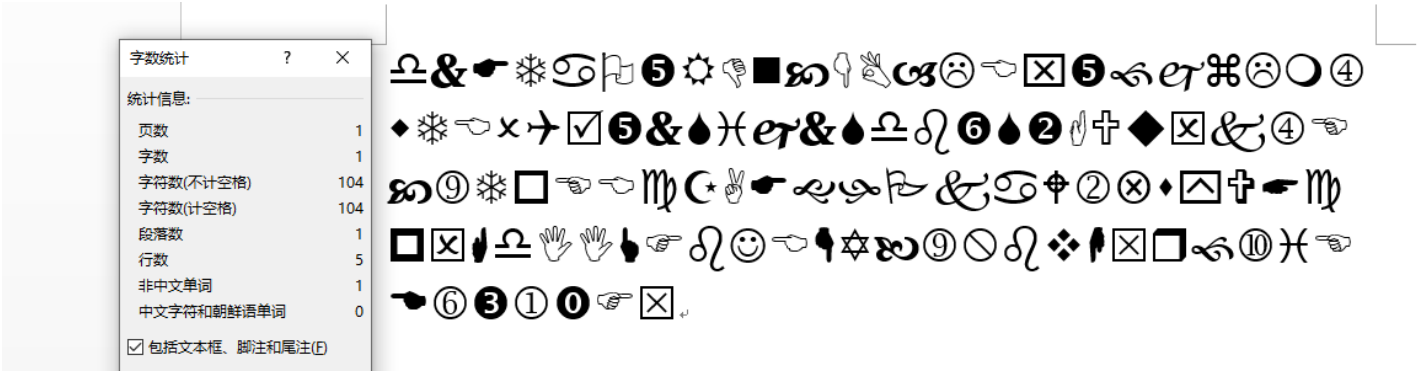


设置一下来压缩，爆破后解压，得到flag.xml文件，里面是第二段flag

之后把zip改成doc，得到这个



字体不一样，同时只有两种字体



将Wingdings的字体视为0，Wingdings2的字体视为1，将其八个分为一组

嗯，特费眼还容易错，而且是动态的flag，别照抄

二进制转字符串的jio本，写太久超时了真是日了狗了，心累

```
01100010
00110101
01100001
00110111
00110001
01100101
01100010
00111000
01100010
01100010
00110111
00110110
01111101
```

```
fp=open("123")
date=fp.read(1000)
n=1
flag=''
num=''
for i in date:
    if i!='\n':
        num+=i
        n+=1
        if n==9:
            flag+=chr(int(num,2))
            num=''
            n=1

print(flag)
```

待续