

Hgame week2 writeup

Web

Cosmos 的新语言

👉 的提示说是写脚本, 但是我研究了半天也不知道从何下手, 目的是什么, 只知道 5s 会变一次, 必须得写脚本, 问了茄子说原来可以直接访问 mycode 这个文件...

Mycode 的内容就是加密一串东西 echo, post 一样的东西就交出 flag.

然后当我一顿研究发现为啥当 base64 解码的时候总是有些字符根本不在 base64 字符的范围里, 又问了茄子才发现 mycode 也会变...

脚本:

```

import base64
import requests
> def dec(str): ...

> def rev(str): ...

> def r13(str): ...

> question = requests.get(...)
firstIndex = question.text.find('</code><br>')

> mycode = requests.get(...)

timu = question.text[firstIndex+12:-21]
print(timu)
code = mycode.text[164:-74]
print(code)
for i in range(10):
>     if (code[0] == 'b'): ...
>     elif (code[0] == 'e'): ...
>     elif (code[0] == 's' and code[3] == 'r'): ...
>     elif (code[0] == 's' and code[3] == '_'): ...

> solutionRequest = requests.post('http://392e390028.php.hgame
flag = solutionRequest.text
print(flag)

```

Re

Unpack

题目说了, 就是手动 upx 脱壳, 按照教程一步一步下来, 得到 bumpfile, 然后 ida 打开, f5, 搜索一下 flag 字符串, 确定了 main 函数位置, 然后发现是简单的加密, flag 加循环变量 i 等于某个值,

自己操作一下, 拿到 flag.

Babypy

打开发现是像类似汇编语言的东西, 然后查了一下 dis 模块

而且还用 00o000 这种东西混淆...

人话形式:

```
def t2y(flag):  
    flag1 = flag[::-1]  
    flag_l = list(flag1)  
    for i in range(1, len(flag_l)):  
        flag_l[i] = flag_l[i - 1] ^ flag_l[i]  
    O = bytes(flag_l)  
  
    return O.hex()
```

解决方法就是倒着异或一边就行, 得到 flag.

Pwn

Findyourself

拖进 ida,f5 之后发现没有任何栈溢出的可能, 那就老老实实做,

可以用的给了两次 system()的机会, 但都有限制,

第一次只能用字母和空格斜杠减号, 在一番 ls 之下, 注意到整个过程只能用到 cat ls sh 和 timeout, timeout 我们明显不用考虑, 第一次调用 system 检查实在严格, 必须要按着题目做等到第二次

system 再“大展身手”，要求知道程序运行目录，翻来翻去只有 /proc 目录下好像能告诉我们什么，于是搜索了一下资料，发现 /proc/self/cwd 会告诉我们程序运行地址。

于是 ls -l /proc/self/ 找到 cwd 旁边就是地址，

输入进地址，然后就到了第二个 system，检测松了一点，但是程序在帮你执行 system()函数之前把 标准输出 关闭了，也就是说你就算想办法 cat flag 了，也看不到。

C 老板提示我先拿到 shell，再把标准输出打开，再拿 flag。

于是用 a=s;b=h;/bin/\$a\$b 躲过 check 检查，拿到 shell

最后一步比较烦，怎么打开标准输出呢，从 /proc/self/fd 里面去找，没有具体在目录里的文件，是直接和 socket 对接的，于是百度到了 2018hctf 的一题，也是关闭了标准输出，解决方法是把标准输出重定向到标准输入(因为标准输入还开着)，然后 cat <空>会打印出输入流里的东西，结合起来 cat flag 就得手了。

Crypto

Verification_code

随机一个 20 位的字符串，让你猜前 4 位，遂爆破。

脚本如下:

```
from socket import socket
from telnetlib import Telnet
from hashlib import sha256
table="1234567890qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM"

sock = socket()
sock.connect(('47.98.192.231', 25678))
sm = sock.recv(1024)
dm = sm[33:len(sm)-1]
sm = sm[12:28]

print(sock.recv(1024))
for a in range(0, 61):
    for b in range(0, 61):
        for c in range(0, 61):
            for d in range(0, 61):
                if (sha256((table[a] + table[b] + table[c] + table[d]).encode() + sm).hexdigest()).encode() == dm:
                    sock.send((table[a]+table[b]+table[c]+table[d]).encode())

sock.send(str.encode("I like playing Hgame"))

print(sock.recv(1024))
print(sock.recv(1024))
print(sock.recv(1024))
```

Misc

Cosmos 的午餐

压缩包解压之后是一个 pcapng 文件, 一个 ssl_log.log 文件, 那就是要通过 ssl_log 的钥匙去读取 pcapng 里的文件, 在编辑->首选项->protocols->TLS(相当与 ssl)加载一下 log 文件.

再筛选 http, 找到一个可疑的文件, winhex 看一下, 发现是 zip, 解压之后是张图片, 名字告诉我们用 outguess 打开, 结合提示: c 老板


喜欢把东西藏在图片备注里, 备注里是一串 key, 用 outguess 打开是一个链接下载下来一张二维码, 扫描二维码拿到 flag.

所见即为假

题目描述大概是“真的是假的, 假的是真的”的意思.

压缩包带密码, 旁边注释 f5 password. 联系题面, 用 7z 打开, 果然是伪压缩, 进去一张 碧蓝航线 的图, 文件名是 FLAG_IN_PICTURE, 再结合题目, 肯定和图片没关系, 这时候 f5 密码就用到了,
java Extract /root/Desktop/FLAG_IN_PICTURE.jpg -p (f5 密码)
得到一个 txt, 里面十六进制, 翻译成文本前面是 rar, 然后说明要把文本形式的 16 进制搞成 2 进制形式, 于是用 winhex 一个一个手打了... 最后得到文件改成 rar 格式解压, 就得到了 flag.

地球上最后的夜晚

打开压缩包, 一个莫名其妙的 pdf 文件名是 no password, 另外一个加密的压缩包, 用  wbStego4.3open.exe 查看 pdf 隐写的东西, 因为说了没密码, 所以真的没有密码, 解出一个 txt 文件, 告诉了我们压缩包的密码, 打开压缩包, 只有一个 word 文档, 仔细找了一下文字间没有隐藏什么, 试试改成 zip, 找到了 secret, 里面是 flag.

玩玩条码

打开是一个 崩坏 3 的 cg 和一个 code128 的压缩包和一个像条码一样的东西.

条码图的文件名是日本的邮政编码, 找了半天没找到能够解码的网站, 只好自己手动一位一位对过去, 得到一串数字密码.

参考资料给了 virtualdub 的东西, 百度到 MSU StegoVideo 隐写视频文件的内容, 用数字密码解开视频文件里隐藏的文件, 得到了 code128 压缩包的密码, 然后 code128 扫一下, 就拿到了 flag.

这周我可太摸鱼了...