VYCTF-hacbit-WP

web

玩蛇（签到） && 玩蛇2.0

直接F12看源码搜索得到flag

玩具沙盒

要输入一段base64，python解码后丢到C验证，先会读4字节作为大写在申请内存，并且要保证读入的大小和申请的一致，由于payload限制长度最多为8，所以直接`\0`绕过，也就是说要传入\0\0\0，对应的base64就是AAAA，即可绕过得到flag

小恐龙

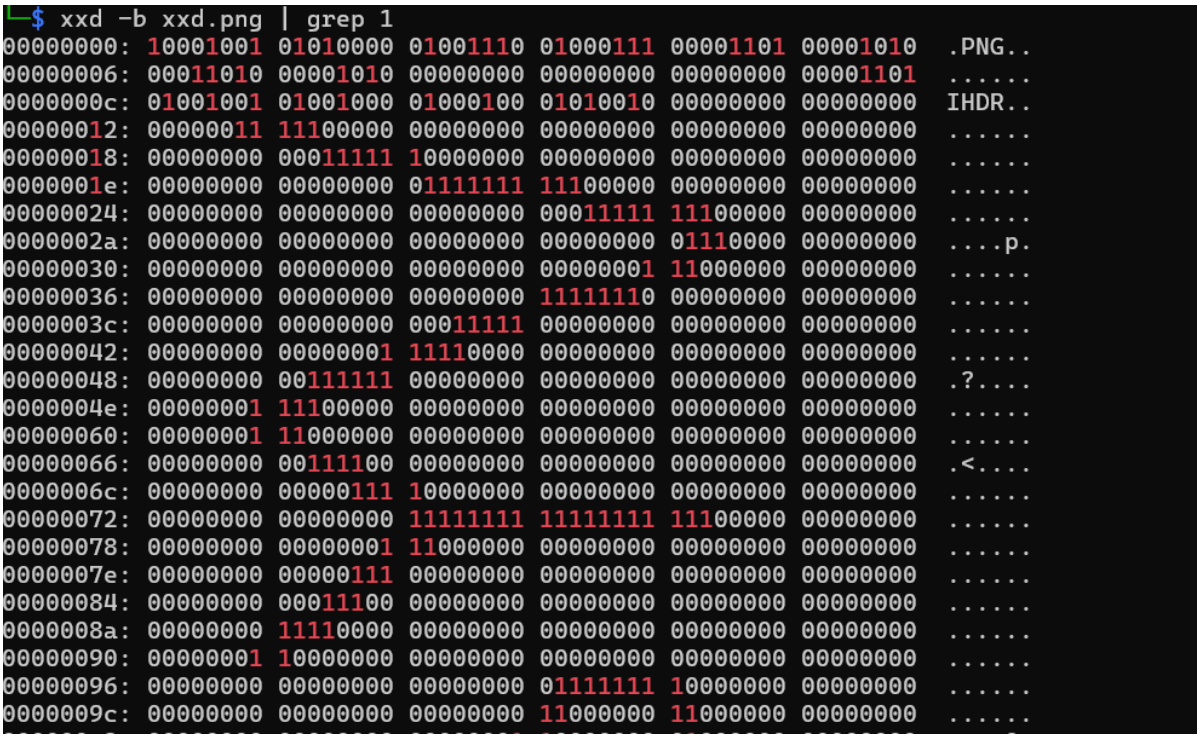
发现可疑的颜色

```
<div class="flag">
                    <li style="color: #89504e"></li>
                    <li style="color: #470d0a"></li>
                    <li style="color: #1a0a00"></li>
                    <li style="color: #00000d"></li>
                    <li style="color: #494844"></li>
                    <li style="color: #520000"></li>
    ......
```

89504e47， 一眼png，把这些数据提取出来保存为png打开是二维码，扫码得到flag

misc

这亦是一种图片



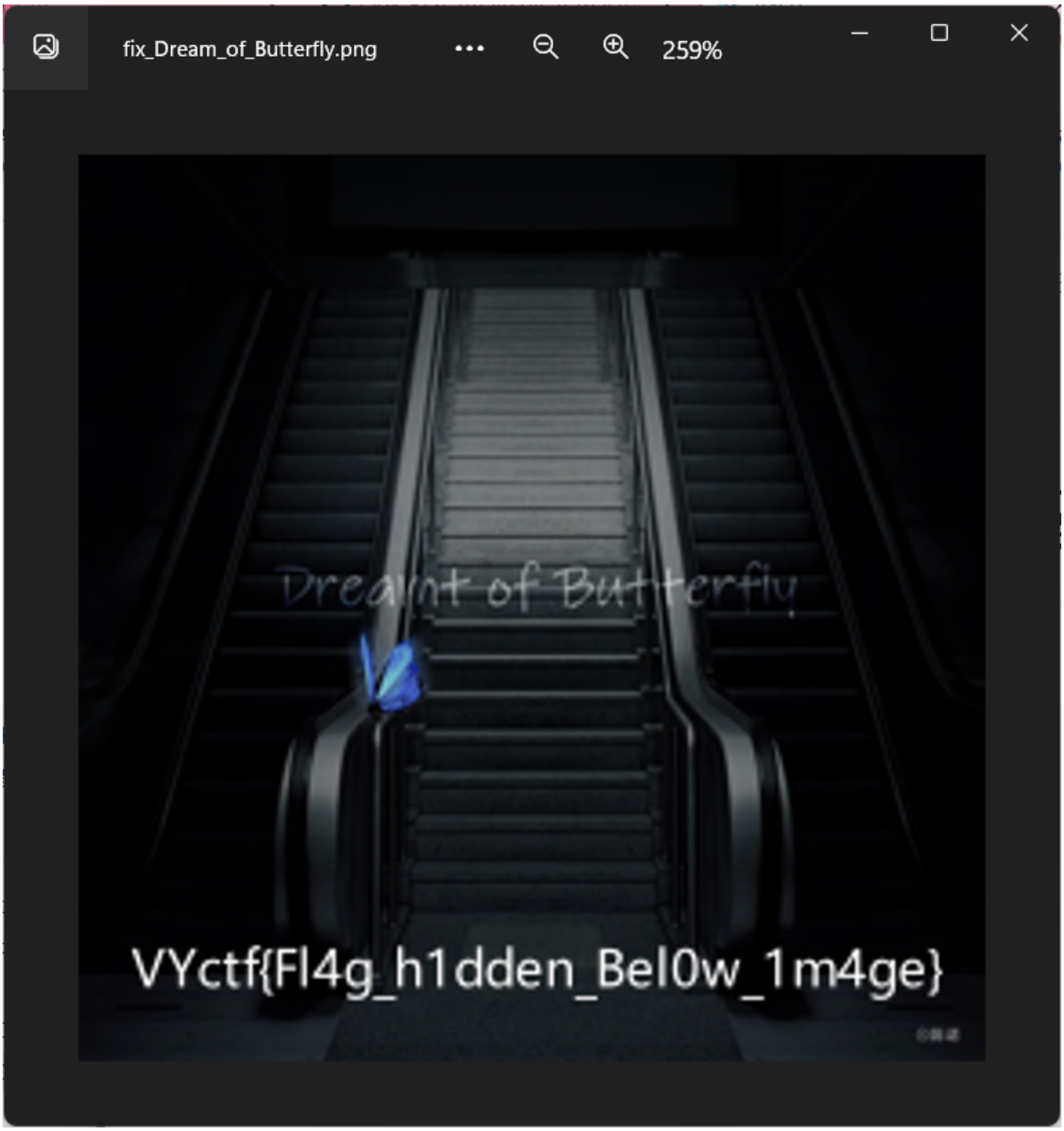
xxd看二进制格式发现怪东西，读出来是`VYctf{Kfc_vw50}`

帕格尼尼的绝望

摁看audacity，结合亿点点的经验和试错，得到mouse，在解摩斯得到`VYCTF/X7BFXXK_DRUM/X7D`，\x7b和\x7d显然是{}， 题目要小写，所以得到flag：`vyctf{fxxk_drum}`

缺少的专辑

puzzlesolver一把梭，修复png得到flag



Rev

大家一起和平地玩耍吧（签到）

摁玩游戏，通关得到flag

base64逆向

把里面那个base64加密的东西解码即可得到flag

`b'vyctf{W31cOm3_7O_vyc7f}'`

二进制

简单逆向，直接复制写c脚本了，py写的麻烦

```c
int main() {
    int v6[41];
    v6[0] = 236;
  v6[1] = 242;
  v6[2] = 198;
  v6[3] = 232;
  v6[4] = 204;
  v6[5] = 246;
  v6[6] = 166;
  v6[7] = 208;
  v6[8] = 216;
  v6[9] = 190;
  v6[10] = 98;
  v6[11] = 230;
  v6[12] = 190;
  v6[13] = 154;
  v6[14] = 96;
  v6[15] = 236;
  v6[16] = 202;
  v6[17] = 190;
  v6[18] = 232;
  v6[19] = 208;
  v6[20] = 202;
  v6[21] = 190;
  v6[22] = 196;
  v6[23] = 98;
  v6[24] = 220;
  v6[25] = 104;
  v6[26] = 228;
  v6[27] = 242;
  v6[28] = 190;
  v6[29] = 232;
  v6[30] = 96;
  v6[31] = 190;
  v6[32] = 232;
  v6[33] = 208;
  v6[34] = 202;
  v6[35] = 190;
  v6[36] = 216;
  v6[37] = 202;
  v6[38] = 204;
  v6[39] = 232;
  v6[40] = 250;
  for (int i = 0; i <= 40; i++) {
    printf("%c", v6[i]>>1);
  }
  return 0;
}
```

`vyctf{Shl_1s_MOve_the_b1n4ry_tO_the_left}`

怎么会解不出来呢

应该是异步或者多线程之类的，执行流和ida直接看到的稍微不太一样，动调即可。动调发现那个异或的东西（懒得回去看ida了），被换成了WWWWWWWW.....，然后就是一个base64换表

```
enc = b'oENJoI1CDOWozzqyo33mOa3sOD/wB6uIzzaZ8H/o8IUoOTwxoLOTBL8iOz4o/EoqAzAxOIUoOxoR'
table = b'abzO12DVop9+/ABtuvEF45678Ocdefgh3mnTUwxyPQRSCijklGHIJKLMNqrsWXYZ'
oldtable = b'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzO123456789+/'
key = b'WWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWWW'

import base64

dec = base64.b64decode(enc.translate(bytes.maketrans(table, oldtable)))
flag = [x^y for x,y in zip(dec, key)]
print(bytes(flag))
# b'vyctf{Oh__yOu_v3_l34rn3d_wh4t_4n_1n1t14l1z3r_funct1On_15}'
```

virus

kawaii病毒

```
fake = b'vyctf{fake_flag}'
flag = [fake[i] ^ fake[15-i] for i in range(16)]
print(bytes(flag))
v17 = 7 + 4*16
flag = [i + (v17:=v17+1) for i in flag]
print(b'VYctf{' + bytes(flag) + b'}')
```

Cry

古老的语言

PS: bf语言不是bf编码，好评

读flag.bf文件，过滤无关字符，把Ow等字符替换为bf语言对应的字符，运行bf代码得到flag

还原大师

ps: 很难评价的一题

肉眼硬瞪flag

```
def function_dict(position):
```

```
    datas = [
            "1sd2jk}3l",
            "wurio456{",
            "8cvn_xm79",
            "etyufgh14",
            "sVhjlaewY",
            "fhjl_ebco",
            "Ysucinowe",
            "Ont5bw_fn"
            ]
    for data in datas:
        assert len(data) == 9,"your dictionary is not quite right."

    return datas[position[0]][position[1]]

def main():
    datas = []
    """ datas.append([function_base64(b"d2VsY29tZQ==",b"welcome"),1])
    datas.append([function_base64(b"dG8=",b"to"),8])
    datas.append([function_base64(b"VlljdGY=",b"VYctf"),4 - 1]) """
    datas = [
            [4, 1], [4, 8], [6, 4-1],# VYc
            [7,3-1], [3,4], [1,9-1], # tf{
            [3,2], [7,0], [1,2-1],  # y0u
            [2,5-1], [5,1-1], [0,0], # _f1
            [2,6-1], [3,0], [0,3-1], # xed
            [5,5-1], [7,3-1], [4,2], # _th
            [3,0], [7,7-1], [0,3-1], # e_d
            [0,0], [2,2-1], [7,3-1], # 1ct
            [0,0], [1,5-1], [2,4-1], # 1on
            [1,6-1], [1,3-1], [3,2], # 4ry
            [0,7-1] # }
    ]
    flag = ""
    for data in datas:
        flag += function_dict(data)
    print(flag)
# VYctf{y0u_f1xed_the_d1ct1on4ry}
```

素数分解

经典rsa

```
E = 7
#N = P * Q
N = 2771
#phin = (P-1) * (Q-1)
#D = pow(E, -1, phin)
# print(D)
D = 1111
""" PT = open("./flag.ct","w")
with open("./flag.pt","r") as file:
    for f in file.read():
```

```
        PT.write(chr((ord(f) ** E) % N))
PT.close()
 """

with open('./flag.ct', 'r', encoding='utf-8') as CT:
    ct = CT.read()

for ch in ct:
    data = (ord(ch) ** D) % N
    print(chr(data), end='')

# vyctf{R5a_1s_MOdern_pA55wOrd}
```

小小的也很可爱哦

经典rsa，不知道该说什么，看脚本的注释吧（）

```
P = 487
# D = ?
E1 = 31
# E2 = pow(E1, D, P)
E2 = 168
R = 11

def enc(PT, E1, E2, R, P):
    C1 = pow(E1, R, P)
    C2 = ""
    for i in PT:
        data = (i * pow(E2, R)) % P
        C2 += chr(data)
    return C1,C2

""" with open("./flag.pt","rb") as PT:
    C1,C2 = enc(PT.read(), E1, E2, R, P)

with open("./flag.ct","w") as CT:
    CT.write("C1 is:"+str(C1)+"\nC2 is:"+C2)

print("C1 is:"+str(C1)+"\nC2 is:"+C2)
 """

#### Decrypt

with open('./flag.ct', 'r', encoding='utf-8') as ct:
    c1, c2 = ct.read().split('\n')
    c1 = int(c1[6:])
    c2 = c2[6:]

print(c1)
print(c2)
# D = 58 + 243 * k
""" for d in range(1, 1000):
    if pow(E1, d, P) == E2:
```

```
        print(d) """

for ch in c2:
    data = (ord(ch) * pow(c1, P-1-58, P)) % P
    print(chr(data), end='')
print()
# VYctf{ElG4m4l_15_4n_45ymmetr1c_encrypt1On_4lgOr1thm}
```

简单sqrt

Permutations(8)也就几万种可能，直接穷举得到2个结果，不过有一个解不出flag（应该就是碰撞了）

注意sagemath重载了^，这个符号代表幂运算而不是异或了，所以要分开写脚本（）

```
'''
P的平方为: [5, 8, 4, 3, 1, 6, 2, 7]
加密结果为: [103, 249, 215, 167, 218, 104, 104, 230, 0, 229, 51, 131, 57, 58, 229, 121, 146, 149, 214, 108, 146, 116, 176, 92, 112, 141, 192, 208, 33, 149, 254, 138, 55, 4, 41, 167, 115, 70]
'''

pall = Premutations(8)
for p in pall:
    if p**2 == [5, 8, 4, 3, 1, 6, 2, 7]:
        print(p)
'''
[3, 7, 5, 1, 4, 6, 8, 2]
[4, 7, 1, 5, 3, 6, 8, 2]
'''
res = [103, 249, 215, 167, 218, 104, 104, 230, 0, 229, 51, 131, 57, 58, 229, 121, 146, 149, 214, 108, 146, 116, 176, 92, 112, 141, 192, 208, 33, 149, 254, 138, 55, 4, 41, 167, 115, 70]
flag = [x^y for x,y in zip(hashlib.sha512(str([3, 7, 5, 1, 4, 6, 8, 2]).encode()).digest(), res)]
# VYctf{We_need_4_MOre_effect1ve_MethOd}
```

简易曲线

看不懂给的表达式要干嘛，把给的数据排序，发现很多重复的，并且基本上相邻的步长为0.7左右，或者接近整数倍，猜测和整数有单一映射关系，并且x加一，y大概加0.7到0.8的样子，把最大的作为 "}", 也就是 0x7d，然后一个个往前推，得到一个映射表，再按照原顺序映射回去，得到flag

```
ct = [51.8, 53.9, 61.1, 73.5, 63.3, 78.6, 41.3, 62.5, 26.2, 68.3, 62.5, 73.5, 72.0, 77.1, 58.2, 71.3, 74.2, 62.5, 29.4, 73.5, 26.8, 69.8, 69.1, 29.4, 58.2, 28.7, 72.0, 62.5, 58.2, 74.2, 29.4, 74.2, 28.7, 67.6, 67.6, 77.1, 58.2, 61.1, 26.2, 68.3, 70.5, 67.6, 62.5, 73.5, 62.5, 61.8, 58.2, 73.5, 64.7, 72.0, 26.2, 74.2, 64.0, 64.7, 58.2, 64.0, 62.5, 26.2, 64.0, 62.5, 60.4, 72.0, 28.7, 80.1]
ct_sort = sorted(ct, reverse=True)
print(ct)
print(ct_sort)
```

```
delta = 0.8
x = [ord('}')]
y = [ct_sort[0]]
i = 0
for ty in ct_sort[1:]:
    i += 1
    if ty in y:
        continue
    if i >= 52:
        delta = 0.7
    oldy = y[-1]
    deltay = oldy - ty
    y.append(ty)
    nx = x[-1] - deltay // delta
    if deltay % delta >= 0.9:
        nx -= 1
    x.append(int(nx - 0.49999999999))
    #print('delta:', delta, 'this y', ty)

print(x)
print(y)
flag = [int(x[y.index(i)]) for i in ct]
print(bytes(flag))
# VYctf{GeOmetry_que5t1on5_4re_u5u4lly_cOmpleted_thrOugh_geOgebr4}
```

注意: 脚本有点问题，改了好多次，每次都差几个字符，最接近的是VYctf{He.....}， 猜测第一个单词是geometry，所以手动把He0metry改成Ge0metry， 提交对了！

建议手搓吧，不好写脚本，写脚本反而慢（）