69：Whitzard OJ逆向（一）

原创

S1lenc3 于 2020-02-07 22:21:10 发布 181 收藏
分类专栏： CTF

> 废物了半个月，不能继续对不起自己了，太菜了，入群题都不会，闭关修炼吧。

simple format

题目名字，简单的格式化，看名字好像是pwn题，IDA看看。

直接看关键处吧。真让人头大，又有新知识可以学了，直接找writeup。

经过参考，弄懂了这一串字符的意思。

%s 决定类型， * 决定宽度， $ 决定第几个参数。

所以 %1$*2$s 就是第一个参数以长度为第二个参数输出，然后发现有x个 %1$*2$s，也就是长度为 x * （第二个参数的值）

其余类似。最后有个%20$n,意思是把输出的字符串的长度放入第二十个参数里。

有18个dprintf，我们输入的字符串的长度是36个，变成word型就是18个。

然后得出是求18个未知数。

```
td,
"%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*"
"2$s%1$*2$s%1$*2$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*5$s%1$*5$s%1$*5$s%1$*5$s%1$*5$s%1$*7$s%"
"1$*7$s%1$*7$s%1$*7$s%1$*7$s%1$*7$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8"
"$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*8$s%1$*9$s%1"
"$*9$s%1$*9$s%1$*9$s%1$*9$s%1$*9$s%1$*9$s%1$*9$s%1$*9$s%1$*9$s%1$*9$s%1$*9$s%1$*9$s%1$*9$s%1$*9$s%1$*9$s%1$*9$s%1$*9$"
"s%1$*9$s%1$*10$s%1$*10$s%1$*10$s%1$*10$s%1$*10$s%1$*10$s%1$*10$s%1$*11$s%1$*11$s%1$*11$s%1$*11$s%1$*11$s%1$*11$s%1$*"
"11$s%1$*11$s%1$*11$s%1$*11$s%1$*11$s%1$*11$s%1$*11$s%1$*11$s%1$*11$s%1$*11$s%1$*11$s%1$*11$s%1$*11$s%1$*11$s%1$*11$s"
"%1$*12$s%1$*12$s%1$*12$s%1$*12$s%1$*12$s%1$*12$s%1$*12$s%1$*12$s%1$*12$s%1$*12$s%1$*12$s%1$*12$s%1$*12$s%1$*12$s%1$*"
"12$s%1$*12$s%1$*14$s%1$*15$s%1$*15$s%1$*15$s%1$*15$s%1$*15$s%1$*15$s%1$*15$s%1$*15$s%1$*15$s%1$*15$s%1$*15$s%1$*15$s"
"%1$*15$s%1$*15$s%1$*15$s%1$*15$s%1$*15$s%1$*15$s%1$*15$s%1$*15$s%1$*15$s%1$*15$s%1$*15$s%1$*16$s%1$*16$s%1$*17$s%1$*"
"17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s"
"%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*17$s%1$*18$s%1$*"
"18$s%1$*18$s%1$*18$s%1$*18$s%1$*18$s%1$*18$s%1$*18$s%1$*18$s%1$*18$s%1$*18$s%1$*18$s%1$*18$s%1$*18$s%1$*19$s%1$*19$s"
"%1$*19$s%1$*19$s%1$*19$s%1$*19$s%1$*19$s%1$*19$s%1$*19$s%1$*19$s%1$*19$s%1$*19$s%1$*19$s%1$*19$s%1$*19$s%1$*19$s%1$*"
"19$s%1$*19$s%1$*19$s%1$*19$s%1$*19$s%20$n",
aAaaaaaaaaaaaaa,
word_6271A0[0],
word_6271A0[1],
word_6271A0[2],
word_6271A0[3],
word_6271A0[4],
word_6271A0[5],
word_6271A0[6],
word_6271A0[7],
word_6271A0[8],
word_6271A0[9],
word_6271A0[10],
word_6271A0[11],
word_6271A0[12],
word_6271A0[13],
word_6271A0[14],
word_6271A0[15],
word_6271A0[16],
word_6271A0[17],
&unk_627200);
```


写了python脚本，发现线性方程组原来这么容易解。。。。

```
s = ["%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$
      "%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$
      "%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$
      "%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$
      "%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$
      "%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$
      "%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$
      "%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$
      "%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$
      "%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*4$s%1$
      "%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$
      "%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$
      "%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*3$s%1$*3$s%1$
      "%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$
      "%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$
      "%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$
      "%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*3$s%1$*3$s%1$
      "%1$*2$s%1$*2$s%1$*2$s%1$*2$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$*3$s%1$
d = [[0] * 18 for i in range(len(s))]
for j in range(len(s)):
    for i in range(2, 20):
        str1 = "%1$*{}$s".format(str(i))
        d[j][i - 2] = s[j].count(str1)
print(d)

for j in range(len(s)):
     f = ''
     for i in range(18):
          f += str(d[j][i]) + chr(ord('a') + i) + '+'
     print(f)

from scipy import linalg
import numpy as np

x = np.array([[19, 8, 0, 5, 0, 6, 27, 19, 6, 22, 16, 0, 1, 23, 2, 29, 14, 22], [26, 18, 9, 14, 3, 1, 0, 6,
y = np.array([5462280,4346506,5891159,6839864,7912833,7049790,7455784,7311612,6299256,7114100,7037043,68730
z = linalg.solve(x, y)
print(z)

d = [21843,21571,31558,12659,28781,13164,28767,26994,14190,24422,12652,25966,29281,26207,29232,30061,24940,
flag = ''
for i in d:
     flag += chr(int('0x' + hex(i)[4:], 16)) + chr(int(hex(i)[:4], 16))
print(flag)
```

参考：https://www.anquanke.com/post/id/146419?mType=Group#h3-8

babyre

这道题拿到两个文件，babyre和out，刚开始没注意out文件，babyre又运行不了。看了看伪代码，没看出来怎么找flag。

主要就是几个函数，

```
seed ^= v3 + (v3 ^ seed);
v2 = (unsigned __int8)v3;
v3 = (unsigned __int8)v3 << 8;
sub_804882B();
sub_8048A41((int)&v4);
sub_80488E0(&v5, 0xA72BE4C1, 0x1D082C23, seed, v3, v2);
sub_8048980(&s);
sub_804882B();
putchar(10);
return 0;
```

都看了一遍，没发现和flag相关的关键词，而且不能运行，很多变量的意义掌握不了，找了一下writeup，发现忽略了out文件。

查看out文件有15行数据，应该是程序输出的数据，整个题输出数据的地方在函数sub_80488e0处。

```
int __cdecl sub_80488E0(void *s, int a2, int a3, int a4, int a5, int a6)
{
  unsigned int v6; // ST4C_4
  signed int i; // [esp+1Ch] [ebp-1Ch]

  memset(s, 0, 0x20u);
  scanf("%s", s);
  for ( i = 0; i <= 29; ++i )
  {
    v6 = sub_804868B(*((char *)s + i), __PAIR__(a3, a2), a4, a6, a5);
    printf("%lx\n", v6);
  }
  return i;
}
```


输出30行，猜测应该是out文件每行可以分为两组。

至此可以发现，只要分析出sub_804868b函数即可。

```
v8 = a1;
for ( j = 0; j <= 527; ++j )
{
  v5 = a2 >> (j & 0x1F);
  if ( j & 0x20 )
    LODWORD(v5) = HIDWORD(v5);
  v8 = (v8 >> 1) ^ (((unsigned int)v5 ^ v8 ^ (v8 >> 16) ^ (0x5C743A2E >> (((v8 >> 1) & 1)
                                                                           + 2
                                                                           * (2
                                                                            * (((v8 >> 20) & 1)
                                                                             + 2
                                                                             * (2 * ((v8 & 0x80000000) != 0)
                                                                              + ((v8 >> 26) & 1)))
                                                                            + ((v8 >> 9) & 1))))) << 31);
}
return v8;
```


经分析，函数只和a1，a2有关。

a1是输入，a2是两个数据的合并，

```
sub_80488E0(&v5, 0xA72BE4C1, 0x1D082C23, seed, v3, v2);
sub_8048980(&s);
```

即a2=0x1D082C23A72BE4C1

直接写脚本，大部分代码可以直接复制伪代码。（参考writeup）

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

uint32_t encrypt(uint32_t a1, uint64_t a2) {
 int j;
 unsigned __int64 v5;
 unsigned int v8;
 v8 = a1;
 for (j = 0; j <= 527; ++j)
 {
  v5 = a2 >> (j & 0x3F);
  v8 = (v8 >> 1) ^ (((unsigned int)v5 ^ v8 ^ (v8 >> 16) ^ (0x5C743A2E >> (((v8 >> 1) & 1)+ 2* (2* (((v8 >>
 }
 return v8;
}
int main() {
 unsigned int data[30] = // out
 {
  0xB80C91FE,0x70573EFE,
  0xBEED92AE,0x7F7A8193,
  0x7390C17B,0x90347C6C,
  0xAA7A15DF,0xAA7A15DF,
  0x526BA076,0x153F1A32,
  0x545C15AD,0x7D8AA463,
  0x526BA076,0xFBCB7AA0,
  0x7D8AA463,0x9C513266,
  0x526BA076,0x6D7DF3E1,
  0xAA7A15DF,0x9C513266,
  0x1EDC3864,0x9323BC07,
  0x7D8AA463,0xFBCB7AA0,
  0x153F1A32,0x526BA076,
  0xF5650025,0xAA7A15DF,
  0x1EDC3864,0xB13AD888
 };
 int i;
 int j;
 for (i = 0; i < 30; i++)
  for (j = 0x20; j < 0x7F; j++)
   if (encrypt(j, 0x1D082C23A72BE4C1) == data[i])
    printf("%c", j);
 printf("\n");
 return 0;
}
```

对v5 = a2 >> (j & 0x3F);的分析：

```
mov     eax, [ebp+var_A4]
and     eax, 3Fh        ; j & 0x3f
mov     ecx, eax
mov     eax, dword ptr [ebp+var_C0]
mov     edx, dword ptr [ebp+var_C0+4]
shrd    eax, edx, cl    ; a2 = edx << 32 + eax
                        ; a2 >> (j & 0x3f)
shr     edx, cl
test    cl, 20h
jz      short loc_8048796
mov     eax, edx
xor     edx, edx
```


shrd ax，bx，2　ax右移两位，空位用bx的低位代替。

逆向过程中的“去伪存真”真的是太重要了。今天还看了一道入群题，压根不知道那个题是个啥意思，难道非得硬怼汇编吗，有空再试试。

参考：https://xz.aliyun.com/t/2347