Writeup(2020.7.8-2020.7.14)

原创

BIAUTUMN 于 2020-07-10 10:27:19 发布 379 收藏 1

文章标签： 信息安全 debug

本文链接：https://blog.csdn.net/BIAUTUMN/article/details/107243941

buuctf新年礼物

查壳，除壳，ida打开

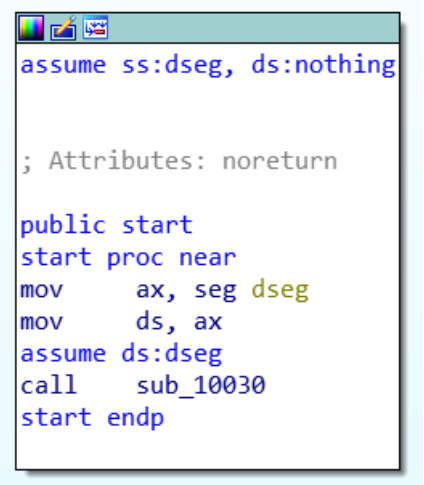
直接找到main函数

观察字符串

flag{HappyNewYear!}

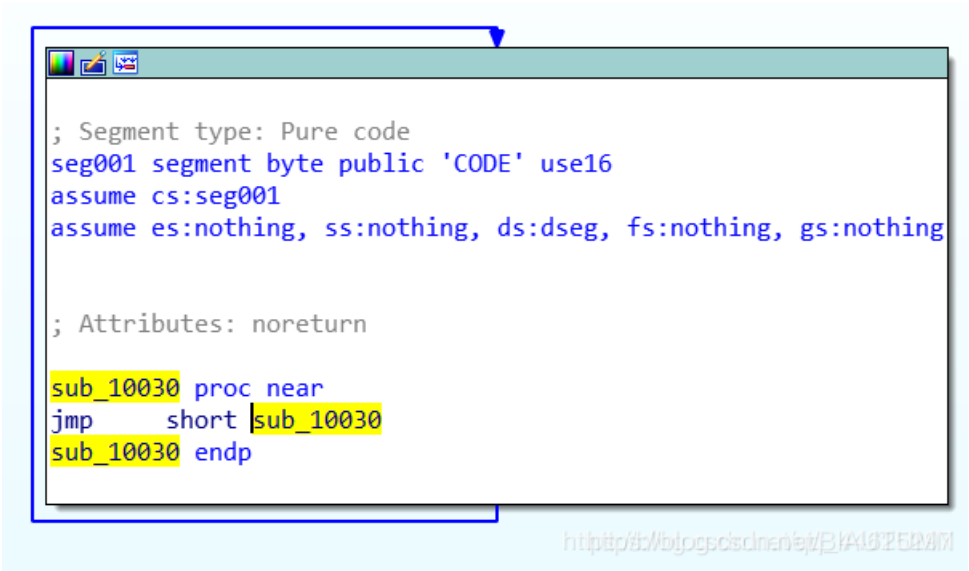
buuctf[BJDCTF 2nd]8086

先查壳，发现没壳，拉进IDA。



Start 函数中调用了sub_10030函数，进去查看

发现sub_10030函数是个死循环，不断跳转到自身。



```
; Segment type: Pure code
seg001 segment byte public 'CODE' use16
assume cs:seg001
assume es:nothing, ss:nothing, ds:dseg, fs:nothing, gs:nothing


; Attributes: noreturn

sub_10030 proc near
jmp     short sub_10030
sub_10030 endp
```

直接查看汇编，这是一个由8086汇编写成的程序。第一段数据段是将这串字符串写入物理地址，并取名aUDuTZWjQGjzZWz。

```
dseg:0000 ; ===========================================================================
dseg:0000
dseg:0000 ; Segment type: Pure data
dseg:0000 dseg            segment para stack 'DATA' use16
dseg:0000                 assume cs:dseg
dseg:0000 aUDuTZWjQGjzZWz db ']U[du~|t@{z@wj.}.~q@gjz{z@wzqW~/b;',0
dseg:0023                 align 10h
dseg:0023 dseg            ends
dseg:0023
seg001:0000 ; ===========================================================================
```

接下来则是代码段，就是刚才的死循环段。但是后面还有一串数据，估计就是真正的代码。选中，按C，选force，强制转换成汇编。

```
seg001:0000 ; ===========================================================================
seg001:0000
seg001:0000 ; Segment type: Pure code
seg001:0000 seg001          segment byte public 'CODE' use16
seg001:0000                 assume cs:seg001
seg001:0000                 assume es:nothing, ss:nothing, ds:dseg, fs:nothing, gs:nothing
seg001:0000
seg001:0000 ; ============== S U B R O U T I N E =======================================
seg001:0000
seg001:0000 ; Attributes: noreturn
seg001:0000
seg001:0000 sub_10030       proc near               ; CODE XREF: sub_10030↓j
seg001:0000                                         ; start+5↓p
seg001:0000                 jmp     short sub_10030
seg001:0000 sub_10030       endp
seg001:0000
seg001:0000 ; ---------------------------------------------------------------------------
seg001:0002                 db 0B9h, 22h, 0, 8Dh, 1Eh, 2 dup(0), 8Bh, 0F9h, 4Fh, 80h
seg001:0002                 db 31h, 1Fh, 0E2h, 0F8h, 8Dh, 16h, 2 dup(0), 0B4h, 9, 0CDh
seg001:0002                 db 21h, 0C3h
seg001:001A                 assume ss:dseg, ds:nothing
seg001:001A
seg001:001A ; ============== S U B R O U T I N E =======================================
```

这里是循环0x22次。字符串和0x1f抑或，抑或的结果就是flag了。

```
a = ']U[du~|t@{z@wj.}.~q@gjz{z@wzqW~/b'
flag = ''
for i in a:
    flag += chr(ord(i)^0x1F)
print(flag)
```

BJD{jack_de_hu1b1an_xuede_henHa0}
此部分借鉴自https://blog.csdn.net/qq_44625297/article/details/105158182

buuctf xor

直接ida打开
查看main函数及其伪代码

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char *v3; // rsi
  int result; // eax
  signed int i; // [rsp+2Ch] [rbp-124h]
  char v6[264]; // [rsp+40h] [rbp-110h]
  __int64 v7; // [rsp+148h] [rbp-8h]

  memset(v6, 0, 0x100uLL);
  v3 = (char *)256;
  printf("Input your flag:\n", 0LL);
  get_line(v6, 256LL);
  if ( strlen(v6) != 33 )
    goto LABEL_12;
  for ( i = 1; i < 33; ++i )
    v6[i] ^= v6[i - 1];
  v3 = global;
  if ( !strncmp(v6, global, 0x21uLL) )
    printf("Success", v3);
  else
LABEL_12:
    printf("Failed", v3);
  result = __stack_chk_guard;
  if ( __stack_chk_guard == v7 )
    result = 0;
  return result;
}
```

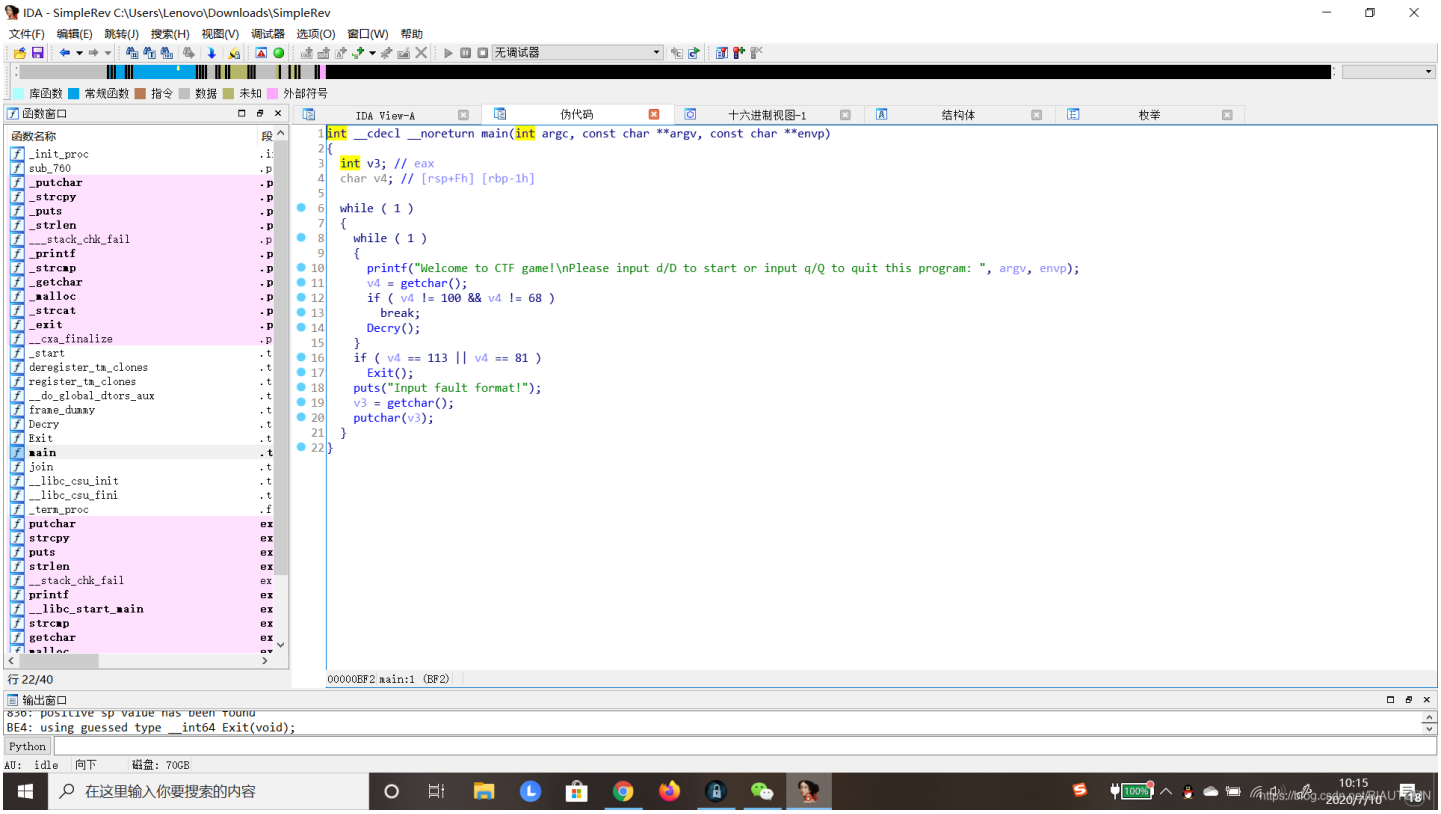
分析可知，flag长度为33位（15,16的循环语句）
并且我们可以得到语句的处理方式，将global反向处理即可
脚本

```
str1 = ['f', 0x0A, 'k', 0x0C, 'w', '&', 'O', '.', '@', 0x11, 'x', 0x0D, 'Z', ';', 'U', 0x11, 'p', 0x19, 'F', 0x1
F, 'v',
        '"', 'M', '#', 'D', 0x0E, 'g', 6, 'h', 0x0F, 'G', '2', 'O']

x = 'f'

for i in range(1, len(str1)):
    if (isinstance(str1[i], str)):
        if (isinstance(str1[i - 1], str)):
            x += chr(ord(str1[i]) ^ ord(str1[i - 1]))
        else:
            x += chr(ord(str1[i]) ^ str1[i - 1])
    else:
        x += chr(str1[i] ^ ord(str1[i - 1]))

print(x)
```

（脚本借鉴自https://www.cnblogs.com/Mayfly-nymph/p/11461575.html）
flag{QianQiuWanDai_YiTongJiangHu}

buuctfSimpleRev

文件直接ida，32位不行就64位
打开找到main函数，f5查看伪代码



发现Detry（）应该是判断函数
打开继续查看
淦，不会了
开始百度
http://shangdixinxi.com/detail-1448227.html

buuctf内涵的软件

查壳，无壳，ida打开
妈的这出题人是智障
真的内涵
一看就看出来了

```
int __cdecl main_0()
{
  int result; // eax
  char v1; // [esp+4Ch] [ebp-Ch]
  const char *v2; // [esp+50h] [ebp-8h]
  int v3; // [esp+54h] [ebp-4h]

  v3 = 5;
  v2 = "DBAPP{49d3c93df25caad81232130f3d2ebfad}";
  while ( v3 >= 0 )
  {
    printf(aD, v3);
    sub_40100A();
    --v3;
  }
  printf(
    "\n"
    "\n"
    "\n"
    "这里本来应该是答案的,但是粗心的程序员忘记把变量写进来了,你要不逆向试试看:(Y/N)\n");
  v1 = 1;
  scanf("%c", &v1);
  if ( v1 == 89 )
  {
    printf(aOdIda);
    result = sub_40100A();
  }
  else
  {
    if ( v1 == 78 )
      printf(asc_425034);
    else
      printf("输入错误,没有提示.");
    result = sub_40100A();
  }
  return result;
}
```

flag就是花括号包的部分

flag{49d3c93df25caad81232130f3d2ebfad}

（持续更新...）