BUU靶场之REVERSE

原创

大佬带带我. 已于 2022-04-02 16:31:09 修改 20 收藏

分类专栏： CTF 逆向 文章标签： c++

于 2022-04-02 16:29:13 首次发布

本文链接：https://blog.csdn.net/qq_39763436/article/details/123917253

今天开始学习CTF中的逆向

大致流程：判断程序是多少位以及是否加了壳，使用IDA进行反编译，在编写脚本

相关工具

【ExeinfoPe】

链接：https://pan.baidu.com/s/16SAvRVydckWYHF3wS93pGQ

提取码：gb3e

【Unpacker_ASPack】

链接：https://pan.baidu.com/s/1m73KxWa-HJgivuvOEb2JFg

提取码：wj33

【UPX Unpacker】

链接：https://pan.baidu.com/s/1QWivSIFFNJuhlwqm3AnZhQ

提取码：er4f

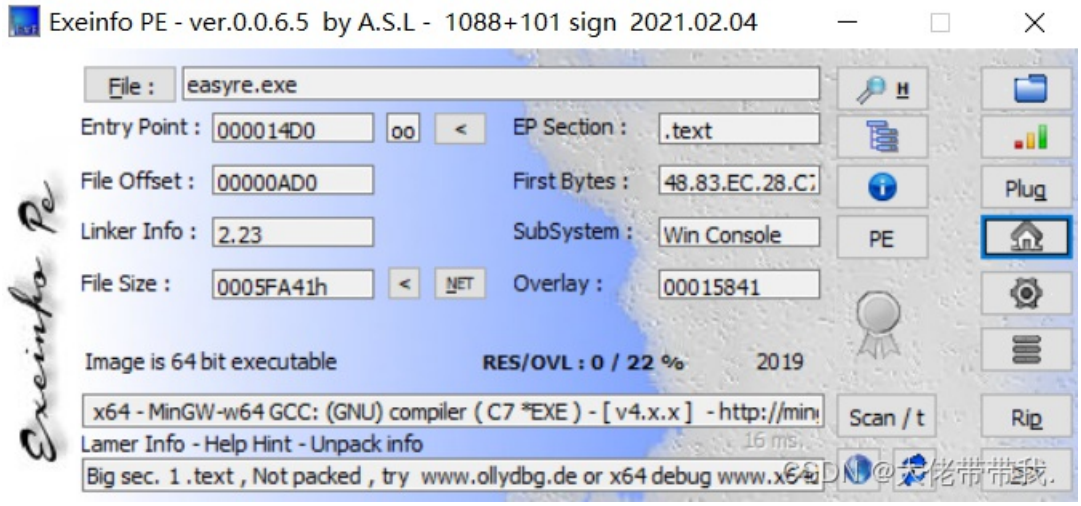
【IDA7.6】

链接：https://pan.baidu.com/s/1VXtb0CB0p11mDmTQlJWtRQ

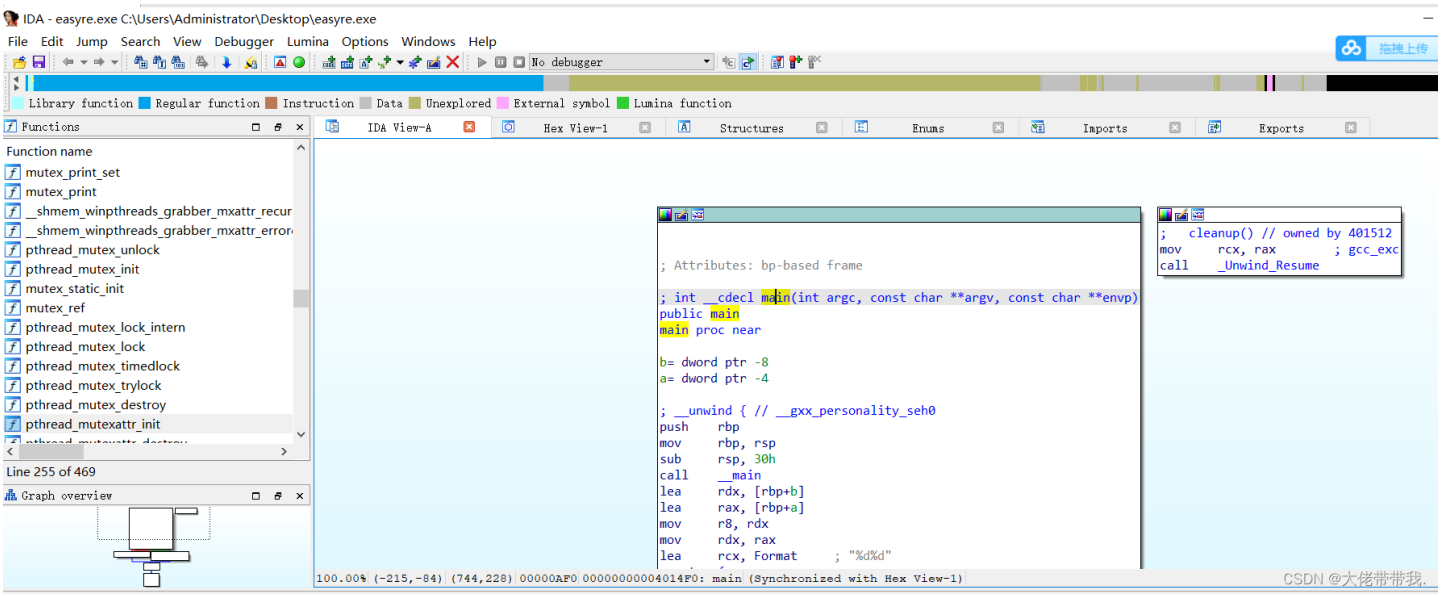
提取码：we3d

easyre

1、下载题目中的压缩包，使用Exeinfo PE工具查看详细信息，发现没加壳



2、使用IDA64查看，找到main()函数，按F5反编译，直接就能看到flag




```c
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int b; // [rsp+28h] [rbp-8h] BYREF
  int a; // [rsp+2Ch] [rbp-4h] BYREF

  _main();
  scanf("%d%d", &a, &b);
  if ( a == b )
    printf("flag{this_Is_a_EaSyRe}");
  else
    printf("sorry,you can't get flag");
  return 0;
}
```


reserse_1

用IDA64打开，找到sub_1400118C0()函数，点击Str2，可以看到Str2是{hello_world}，把o替换成0，那么flag为{hell0_w0rld}

```
.data:000000014001C000 ; char Str2[]
.data:000000014001C000 Str2            db '{hello_world}',0    ; DATA XREF: sub_1400118C0+4B↑o
.data:000000014001C000                                         ; sub_1400118C0+67↑o ...
```

```
__int64 v7; // [rsp+128h] [rbp+108h]

v0 = v4;
for ( i = 82i64; i; --i )
{
  *(_DWORD *)v0 = -858993460;
  v0 += 4;
}
for ( j = 0; ; ++j )
{
  v7 = j;
  if ( j > j_strlen(Str2) )
    break;
  if ( Str2[j] == 111 )
    Str2[j] = 48;
}
sub_1400111D1("input the flag:");
sub_14001128F("%20s", Str1);
v2 = j_strlen(Str2);
if ( !strncmp(Str1, Str2, v2) )
  sub_1400111D1("this is the right flag!\n");
else
  sub_1400111D1("wrong flag\n");
sub_14001113B(v4, &unk_140019D00);
return 0i64;
}
```



```
    break;
  if ( Str2[j] == 'o' )
    Str2[j] = '0';
}
```

reserse_2

打开IDA64，找到main()函数，F5反编译，s2就是输入的flag，由代码可知flag=s2，for循环中的flag由于之前没有定义过，说明它是由值得，双击flag就可以看到它的值，for循环里面的代码作用是用1替换i和r，所以flag为{hack1ng_fo1_fun}

IDA View-A | Pseudocode-A | Hex View-1 | Str

```
4  int i; // [rsp+8h] [rbp-38h]
5  __pid_t pid; // [rsp+Ch] [rbp-34h]
6  char s2[24]; // [rsp+10h] [rbp-30h] BYREF
7  unsigned __int64 v8; // [rsp+28h] [rbp-18h]
8
9  v8 = __readfsqword(0x28u);
10 pid = fork();
11 if ( pid )
12 {
13   waitpid(pid, &stat_loc, 0);
14 }
15 else
16 {
17   for ( i = 0; i <= strlen(&flag); ++i )
18   {
19     if ( *(&flag + i) == 'i' || *(&flag + i) == 'r' )
20       *(&flag + i) = '1';
21   }
22 }
23 printf("input the flag:");
24 __isoc99_scanf("%20s", s2);
25 if ( !strcmp(&flag, s2) )
26   return puts("this is the right flag!");
27 else
28   return puts("wrong flag!");
29 }
```



```
.data:0000000000601080 ; char flag
.data:0000000000601080 flag            db '{'                  ; DATA XREF: main+34↑r
.data:0000000000601080                                         ; main+44↑r ...
.data:0000000000601081 aHackingForFun  db 'hacking_for_fun}',0
.data:0000000000601081 _data           ends
.data:0000000000601081
```

内涵的软件

用IDA32打开，找到main()函数，F5反编译，点击main_0，直接能看到类似flag的字符串，提交试了一下直接成功了，flag{49d3c93df25caad81232130f3d2ebfad}

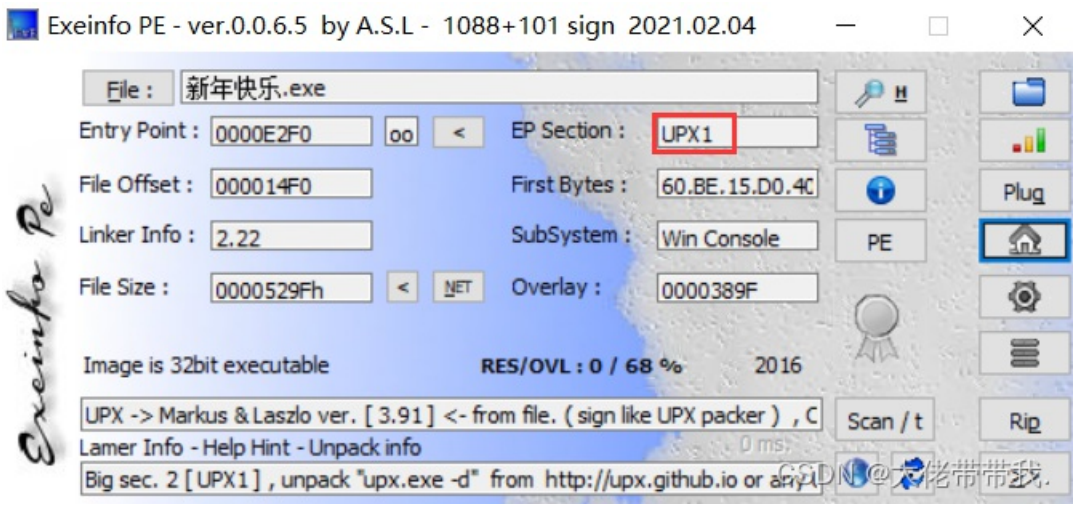
```
// attributes: thunk
int __cdecl main(int argc, const char **argv, const char **envp)
{
  return main_0(argc, argv, envp);
}
```

```
int __cdecl main_0(int argc, const char **argv, const char **envp)
{
  char v4[4]; // [esp+4Ch] [ebp-Ch] BYREF
  const char *v5; // [esp+50h] [ebp-8h]
  int v6; // [esp+54h] [ebp-4h]

  v6 = 5;
  v5 = "DBAPP{49d3c93df25caad81232130f3d2ebfad}";
  while ( v6 >= 0 )
  {
    printf(&byte_4250EC, v6);
    sub_40100A();
    --v6;
  }
  printf(asc_425088);
  v4[0] = 1;
  scanf("%c", v4);
  if ( v4[0] == 89 )
```


新年快乐

1、使用Exeinfo PE工具看到加了UPX1的壳，使用UPX Unpacker工具进行脱壳



2、用IDA32打开脱壳后的程序，找到main()函数，F5反编译，flag=Str2=HappyNewYear!

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char Str2[14]; // [esp+12h] [ebp-3Ah] BYREF
  char flag[44]; // [esp+20h] [ebp-2Ch] BYREF

  sub_401910();
  strcpy(Str2, "HappyNewYear!");
  memset(flag, 0, 32);
  printf("please input the true flag:");
  scanf("%s", flag);
  if ( !strncmp(flag, Str2, strlen(Str2)) )
    return puts("this is true flag!");
  else
    return puts("wrong!");
}
```


xor

1、用IDA64打开，找到main()函数，F5反编译

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int i; // [rsp+2Ch] [rbp-124h]
  char flag[264]; // [rsp+40h] [rbp-110h] BYREF

  memset(flag, 0, 0x100uLL);
  printf("Input your flag:\n");
  get_line(flag, 256LL);
  if ( strlen(flag) != 33 )
    goto LABEL_7;
  for ( i = 1; i < 33; ++i )
    flag[i] ^= flag[i - 1];
  if ( !strncmp(flag, global, 0x21uLL) )
    printf("Success");
  else
LABEL_7:
    printf("Failed");
  return 0;
}
```


2、双击查看global(右键，点DATA)，一共有32位

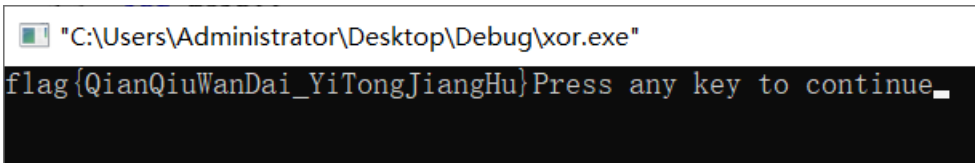
```
__cstring:0000000100000F6E                 ;org 100000F6Eh
__cstring:0000000100000F6E aFKWOXZUPFVMDGH db 'f',0Ah              ; DATA XREF: __data:_global↓o
__cstring:0000000100000F6E                 db 'k',0Ch,'w&O.@',11h,'x',0Dh,'Z;U',11h,'p',19h,'F',1Fh,'v"M#D',0Eh,'g'
__cstring:0000000100000F6E                 db 6,'h',0Fh,'G2O',0
__cstring:0000000100000F90 aInputYourFlag  db 'Input your flag:',0Ah,0
__cstring:0000000100000F90                                         ; DATA XREF: _main+B↑o
```

```
__cstring:0000000100000F6E byte_100000F6E  db 66h                  ; DATA XREF: __data:_global↓o
__cstring:0000000100000F6F                 db  0Ah
__cstring:0000000100000F70                 db  6Bh ; k
__cstring:0000000100000F71                 db  0Ch
__cstring:0000000100000F72                 db  77h ; w
__cstring:0000000100000F73                 db  26h ; &
__cstring:0000000100000F74                 db  4Fh ; O
__cstring:0000000100000F75                 db  2Eh ; .
__cstring:0000000100000F76                 db  40h ; @
__cstring:0000000100000F77                 db  11h
__cstring:0000000100000F78                 db  78h ; x
__cstring:0000000100000F79                 db  0Dh
__cstring:0000000100000F7A                 db  5Ah ; Z
__cstring:0000000100000F7B                 db  3Bh ; ;
__cstring:0000000100000F7C                 db  55h ; U
__cstring:0000000100000F7D                 db  11h
__cstring:0000000100000F7E                 db  70h ; p
__cstring:0000000100000F7F                 db  19h
__cstring:0000000100000F80                 db  46h ; F
__cstring:0000000100000F81                 db  1Fh
__cstring:0000000100000F82                 db  76h ; v
__cstring:0000000100000F83                 db  22h ; "
__cstring:0000000100000F84                 db  4Dh ; M
__cstring:0000000100000F85                 db  23h ; #
__cstring:0000000100000F86                 db  44h ; D
```


3、在根据逻辑写个脚本xor.c

```
#include<stdio.h>

int main()
{
 int flag[33] = {0x66,0x0A,0x6B,0x0C,0x77,0x26,0x4F,0x2E,0x40,0x11,0x78,0x0D,0x5A,0x3B,0x55,0x11,0x70,0x19,0x46,
0x1F,0x76,0x22,0x4D,0x23,0x44,0x0E,0x67,0x6,0x68,0x0F,0x47,0x32,0x4F};
 int i;
 for (i = 32; i > 0; --i )
 {
  flag[i] ^= flag[i - 1];
 }
    for(i = 0; i < 33; i++)
 {
  printf("%c",flag[i]);
 }
 return 0;
}
```



[ACTF新生赛2020]SoulLike

1、用IDA64打开，找到main()函数，F5反编译flag前10位为actf{actf{

```
  v10 = __readfsqword(0x28u);
  printf("input flag:");
  scanf("%s", &flag[6]);
  strcpy(flag, "actf{");
  v5 = 1;
  for ( i = 0; i <= 4; ++i )
  {
    if ( flag[i] != flag[i + 6] )
    {
      v5 = 0;
      goto LABEL_6;
    }
  }
```


2、这段代码是计算flag的11位到22位，且等于v8

```
LABEL_6:
  for ( j = 0; j <= 11; ++j )
    v8[j] = flag[j + 11];
  if ( (unsigned __int8)sub_83A(v8) && flag[23] == '}' )
  {
    printf("That's true! flag is %s", &flag[6]);
    return 0LL;
  }
  else
```

3、查看一下sub_83A()函数，进行了一系列的运算，v8运算的最后结果就等于v3【每次运算得到的值，在进行下一位运算时会使用到】

```
__int64 __fastcall sub_83A(_DWORD *a1)
{
  int i; // [rsp+1Ch] [rbp-44h]
  int v3[14]; // [rsp+20h] [rbp-40h]
  unsigned __int64 v4; // [rsp+58h] [rbp-8h]

  v4 = __readfsqword(0x28u);
  *a1 ^= 0x2Bu;
  a1[1] ^= 0x6Cu;
  a1[2] ^= 0x7Eu;
  a1[3] ^= 0x56u;
  a1[4] ^= 0x39u;
  a1[5] ^= 3u;
  a1[6] ^= 0x2Du;
  a1[7] ^= 0x28u;
  a1[8] ^= 8u;
  ++a1[9];
  a1[10] ^= 0x2Fu;
  a1[11] ^= 0xAu;
  ++*a1;
  a1[1] ^= 0xDu;
  a1[2] ^= 0x73u;
  a1[3] ^= a1[2];
  a1[4] ^= 0x37u;
  ++a1[5];
  a1[6] ^= 0x69u;
```



```
  a1[11] ^= 0x3Bu;
  v3[0] = 126;
  v3[1] = 50;
  v3[2] = 37;
  v3[3] = 88;
  v3[4] = 89;
  v3[5] = 107;
  v3[6] = 53;
  v3[7] = 110;
  v3[8] = 0;
  v3[9] = 19;
  v3[10] = 30;
  v3[11] = 56;
  for ( i = 0; i <= 11; ++i )
  {
    if ( v3[i] != a1[i] )
    {
      printf("wrong on #%d\n", (unsigned int)i);
      return 0LL;
    }
  }
  return 1LL;
}
```


4、编写脚本

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

int main()
{
 char a1[13] = "", temp[13] = "";
 char v3[] = {126,50,37,88,89,107,53,110,0,19,30,56};
 int i,j;
    for(i=0; i<12; i++)
 {

  for(j=0; j<255; j++)
  {
   strcpy(a1, temp);
   a1[i] = j;
   *a1 ^= 0x2Bu;
   a1[1] ^= 0x6Cu;
   a1[2] ^= 0x7Eu;
   .....
   a1[11] ^= 0x3Bu;

   if(a1[i] == v3[i])
   {
    temp[i] = j;
    break;
   }
  }
 }
 printf("flag{%s}", temp);
 return 0;
}
```