攻防世界re1 writeup

原创

Hush^ 于 2020-11-18 10:39:08 发布 197 收藏
分类专栏： 新手 逆向

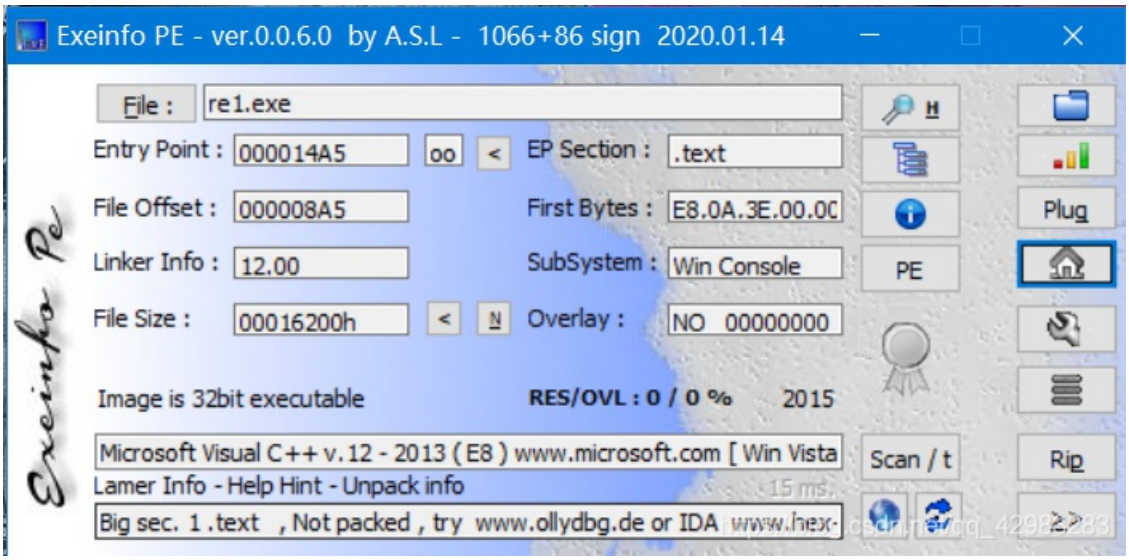
本文链接：https://blog.csdn.net/qq_51687381/article/details/109765924

新手 同时被 2 个专栏收录
21 篇文章 0 订阅
订阅专栏

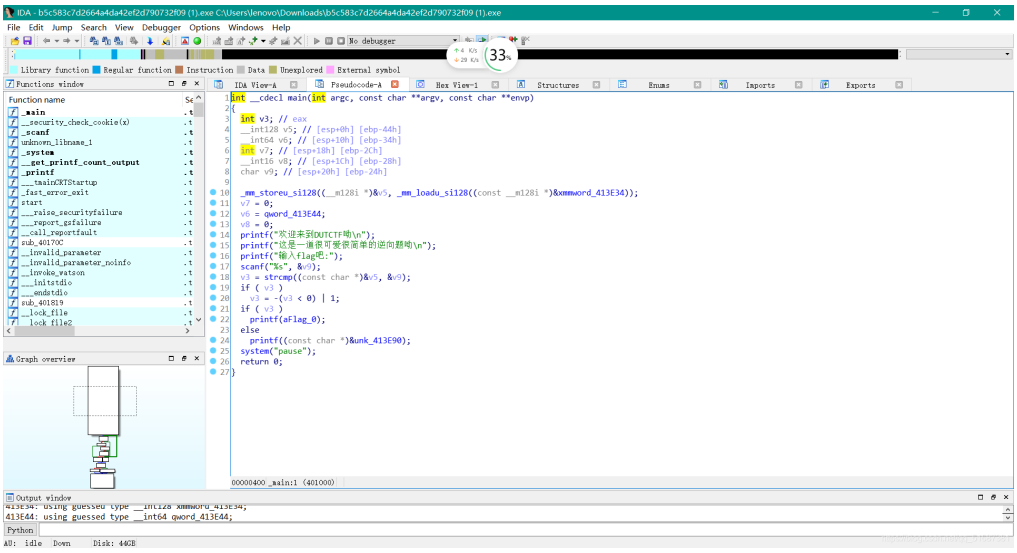
逆向
1 篇文章 0 订阅
订阅专栏

初入ctf，尝试做了一道题，深有体会，写一篇博客记录一下！

1 查壳 ： 无壳，32位



2;ida打开程序，找到main函数，按下F5反汇编；



3；阅读一下发现是对比v5，v9，若v5相等，即v3=0则进入flag get。

```
else
  printf((const char *)&unk_413E90);
system("pause");
return 0;
```

```
.rdata:00413E90 unk_413E90      db  66h ; f
.rdata:00413E91                 db  6Ch ; l
.rdata:00413E92                 db  61h ; a
.rdata:00413E93                 db  67h ; g
.rdata:00413E94                 db  20h
.rdata:00413E95                 db  67h ; g
.rdata:00413E96                 db  65h ; e
.rdata:00413E97                 db  74h ; t
```

若不等于0，则v3会进行运算使v3！=0，得到

```
0413E9C aFlag_0         db 'flag不太对呦'
```

ps：

```
_mm_storeu_si128((__m128i *)&v5, _mm_loadu_si128((const __m128i *)&xmmword_413E34));
```

（给自己记个笔记）

1：_mm_storeu_si128 (__m128i *p, __m128i a);

指令名：void **_mm_storeu_si128 (__m128i *p, __m128i a);**

功能：可存储128位数据；

说明：将__m128i 变量a的值存储到p所指定的变量中去；

注意：p不要求必须是一个16-bit对齐的一个变量的地址。

2：

__m128i _mm_loadu_si128 (__m128i *p);

指令名：__m128i _mm_load_si128 (__m128i *p);

说明：加载128bits值；

返回值：返回可以存放在代表寄存器的变量中的值；

注意：p不用是一个16-bit对齐的一个变量的地址；

（万能的csdn）

4：找到xmmword_413E34

```
4 xmmword_413E34  xmmword 3074656D30633165577B465443545544h
```

是一串16进制数据，我们可以将他转化为字符串就可以得到flag；

方法一：直接选中按下R键

```
xmmword_413E34  xmmword '0tem0c1eW{FTCTUD'
                                        ; [
qword_413E44    dq '}FTCTUD'            ; [
```

ps：

在x86处理器中，数据采用小端序储存；

即高位字节在高位地址，低位字节在地位地址；

方法二：百度，csdn转换代码