攻防世界逆向入门题之simple-unpack

原创

沐一 · 林 于 2021-08-04 22:56:38 发布 130 收藏

分类专栏： CTF 逆向 文章标签： unctf

本文链接：https://blog.csdn.net/xiao__1bai/article/details/119395006

CTF 同时被 2 个专栏收录

167 篇文章 6 订阅

订阅专栏

逆向

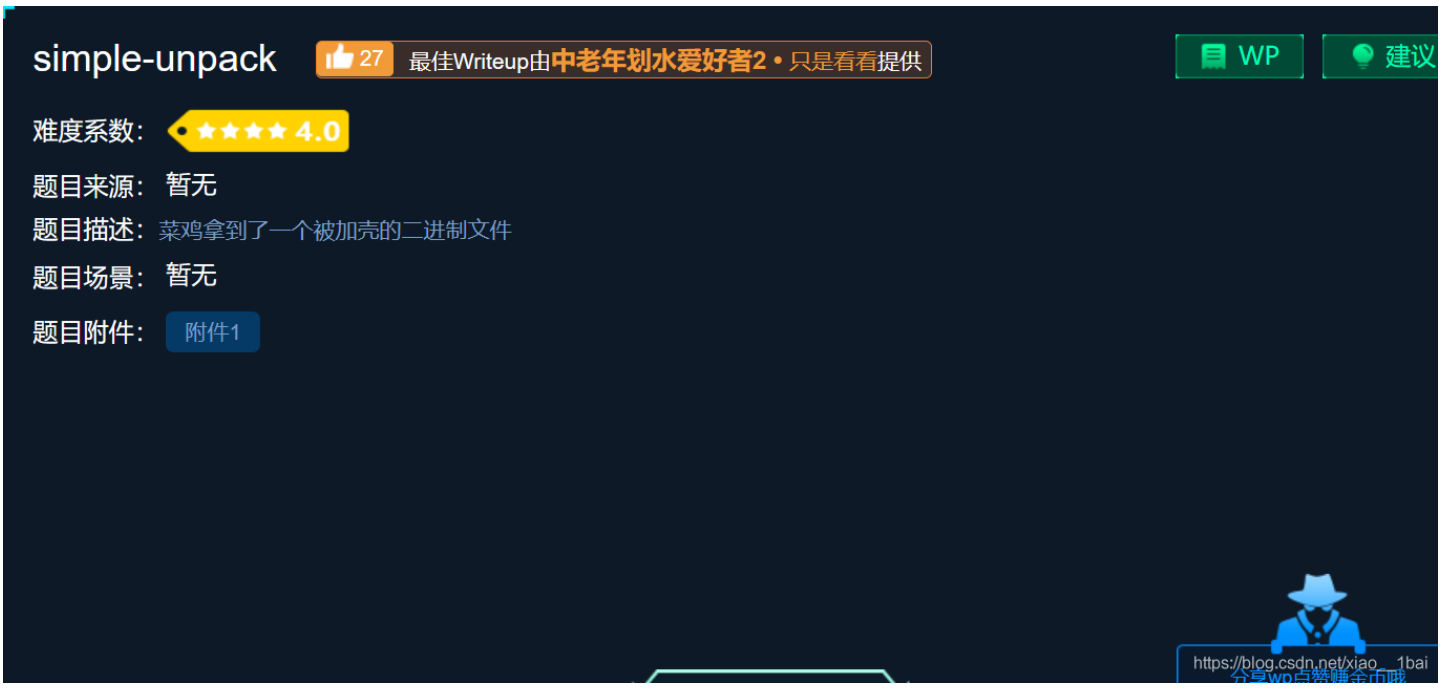
95 篇文章 6 订阅

订阅专栏

攻防世界逆向入门题之simple-unpack

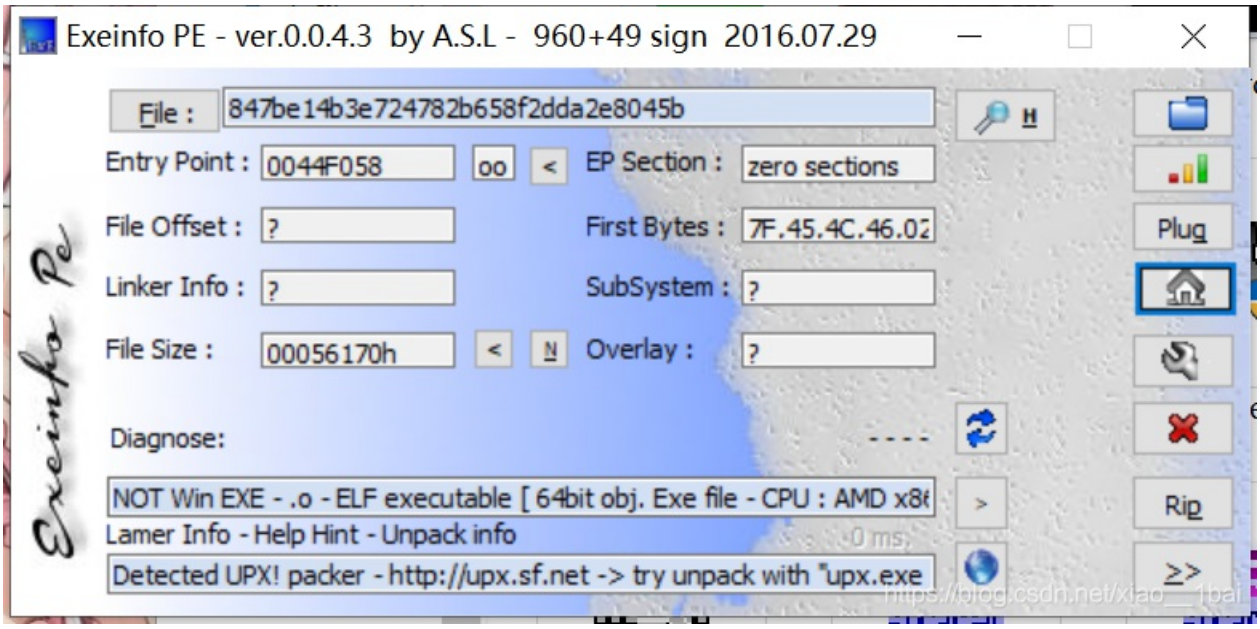
继续开启全栈梦想之逆向之旅~

这题是攻防世界逆向入门题之simple-unpack



这是我第一次接触加壳的题，照着套路扔到PEID中查看信息

无果，想起可能是linux的ELF可执行文件，扔到exeinfo中，



显示说探测到UPX壳，由于第一次做带壳的题目，所以查到了以下资料：

UPX (the Ultimate Packer for eXecutables)是一款先进的可执行程序文件压缩器，压缩过的可执行文件体积缩小50%-70%，这样减少了磁盘占用空间、网络上传下载的时间和其它分布以及存储费用。 通过 UPX 压缩过的程序和程序库完全没有功能损失和压缩之前一样可正常地运行，对于支持的大多数格式没有运行时间或内存的不利后果。 UPX 支持许多不同的可执行文件格式 包含 Windows 95/98/ME/NT/2000/XP/CE 程序和动态链接库、DOS 程序、 Linux 可执行文件和核心。

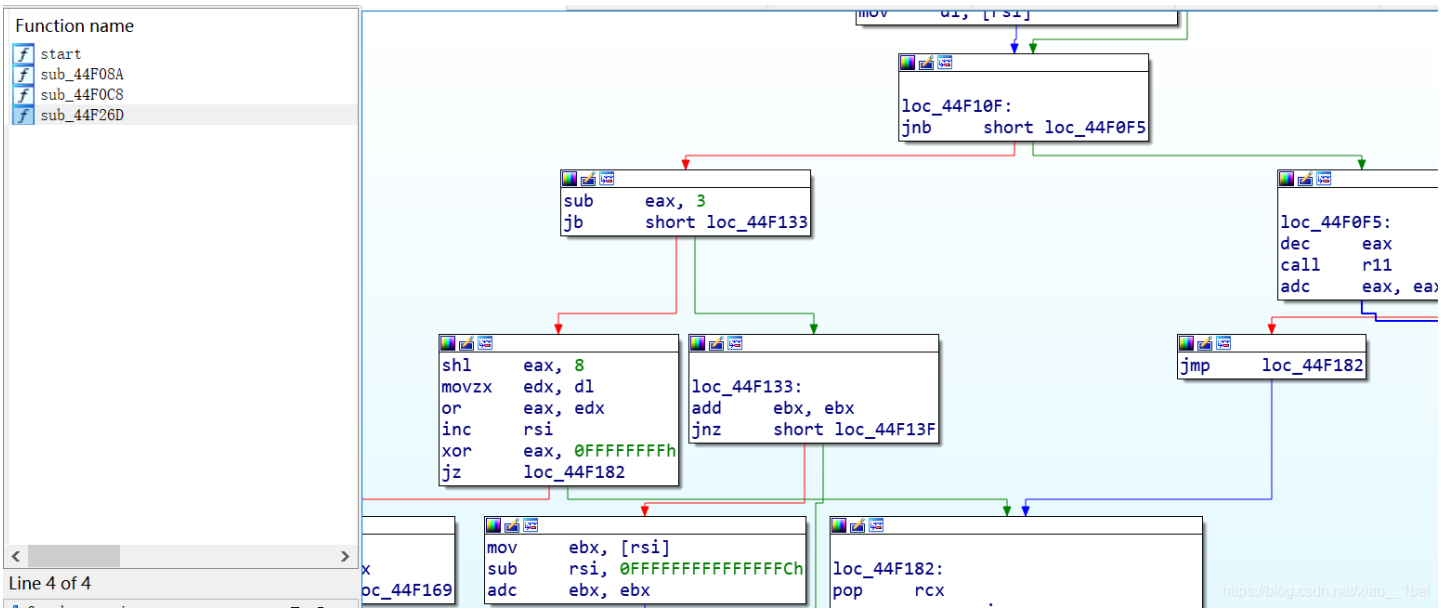
UPX是一个压缩工具，好在今天准备看《逆向核心工程原理》这本书的压缩部分，原来这就是压缩，之前也学了一点PE文件格式，知道了一些文件资源的存放位置，那么下一步就是脱壳了，查到了kali中关于UPX的脱壳命令：upx -d filename
脱完壳就可以直接IDA查看了，FLAG直接就显示出来了

```
main proc near

s1= byte ptr -70h
var_8= qword ptr -8

; __unwind {
push    rbp
mov     rbp, rsp
sub     rsp, 70h
mov     rax, fs:28h
mov     [rbp+var_8], rax
xor     eax, eax
lea     rax, [rbp+s1]
mov     rsi, rax
mov     edi, offset a96s ; "%96s"
mov     eax, 0
call    __isoc99_scanf
lea     rax, [rbp+s1]
mov     esi, offset flag ; "flag{Upx_1s_n0t_a_d3liv3r_c0mp4ny}"
mov     rdi, rax        ; s1
call    _strcmp
test    eax, eax
jnz     short loc_4009FC
```

加壳怎么说，以我目前大致的理解是对文件格式的操作，IDA等二进制分析器应该都需要完整的文件格式才能分析，压缩后(加壳)的文件由于并没有破坏可执行文件的格式规则，所以还是可以运行的，用IDA分析加壳后的文件就分析不出来了，如图：



少得可怜。