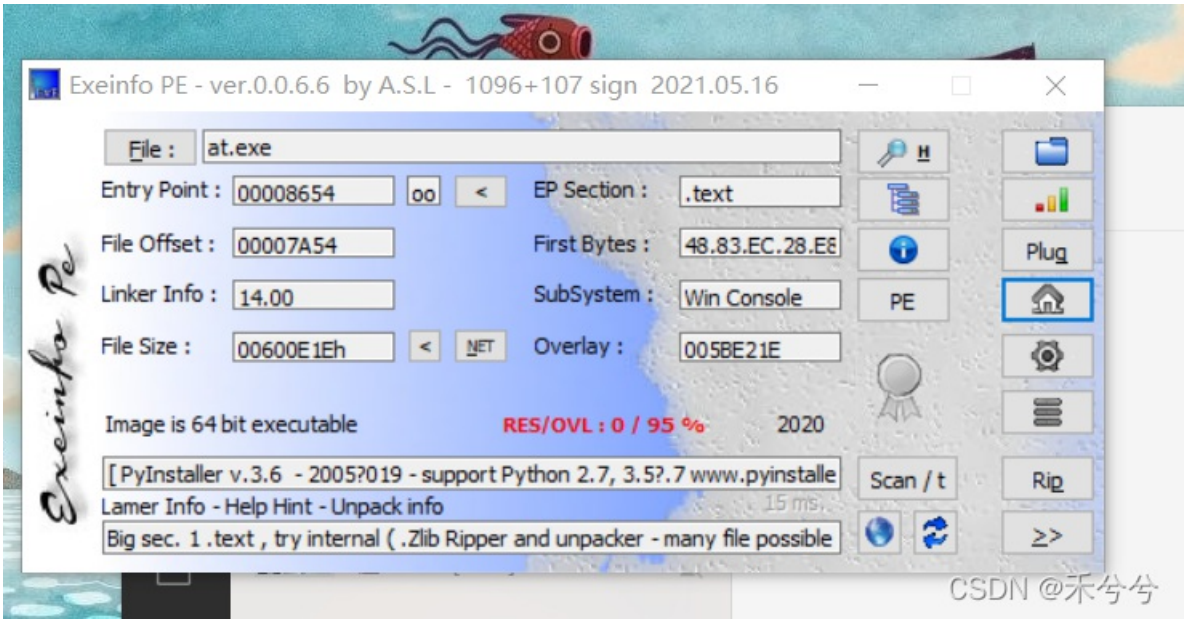
# [羊城杯 2020]login writeup

原创

禾兮兮 于 2021-12-29 22:20:09 发布 401 收藏
文章标签： 其他 经验分享

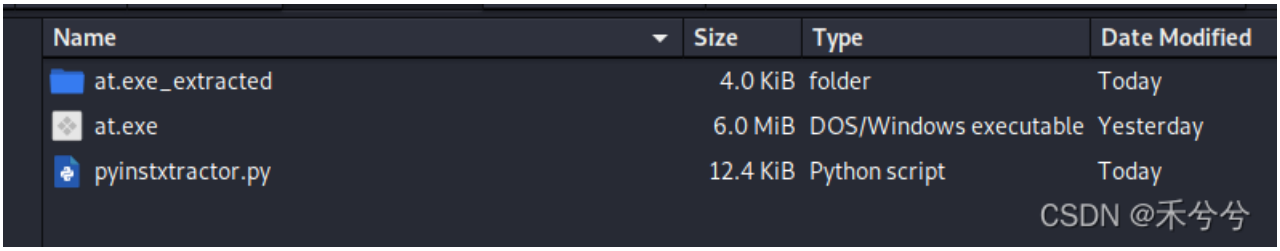

ExeinfoPE查壳，之前我用0.3的版本查不出来，因为没有壳，打开发现什么都找不到，更新了ExeinfoPE才找到了pyinstaller的壳。



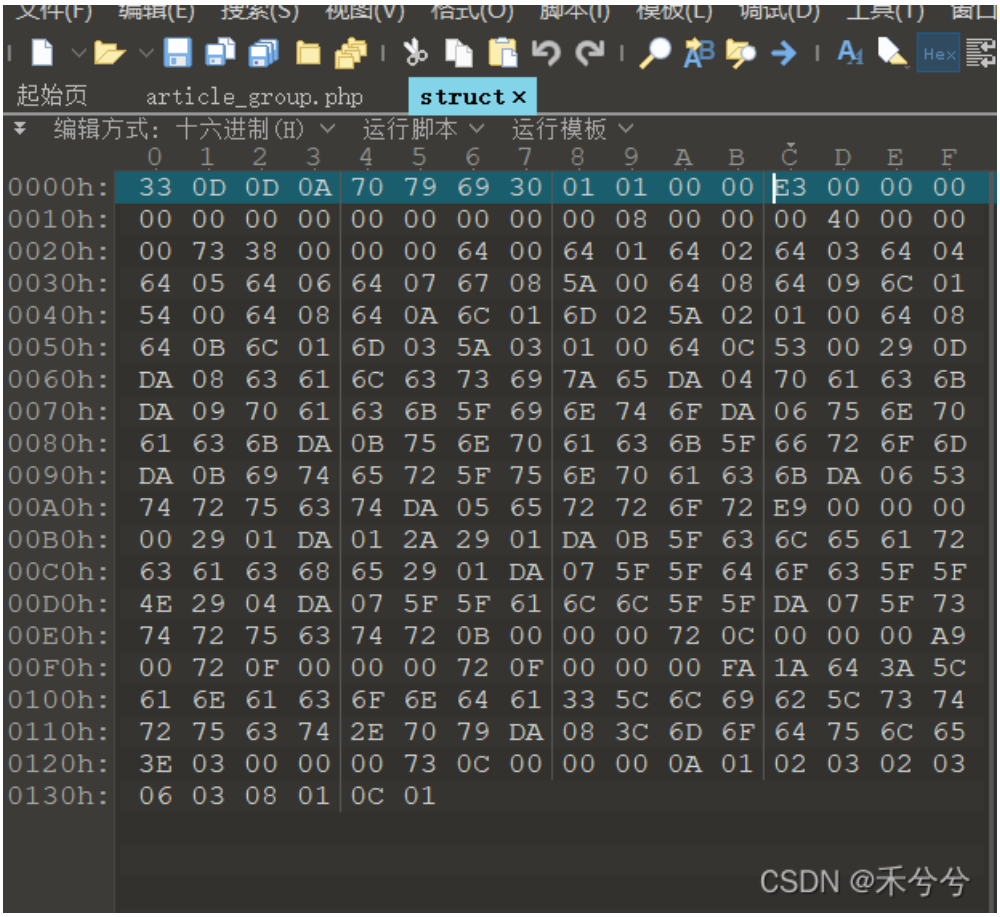

至于pyinstaller，翻了很多文章，找到了它的打包原理，原来是利用Cython把Python转换成C语言，然后编译打包。(43条消息) Pyinstaller原理详解_XHG78999的代码之家-CSDN博客_pyinstaller讲解

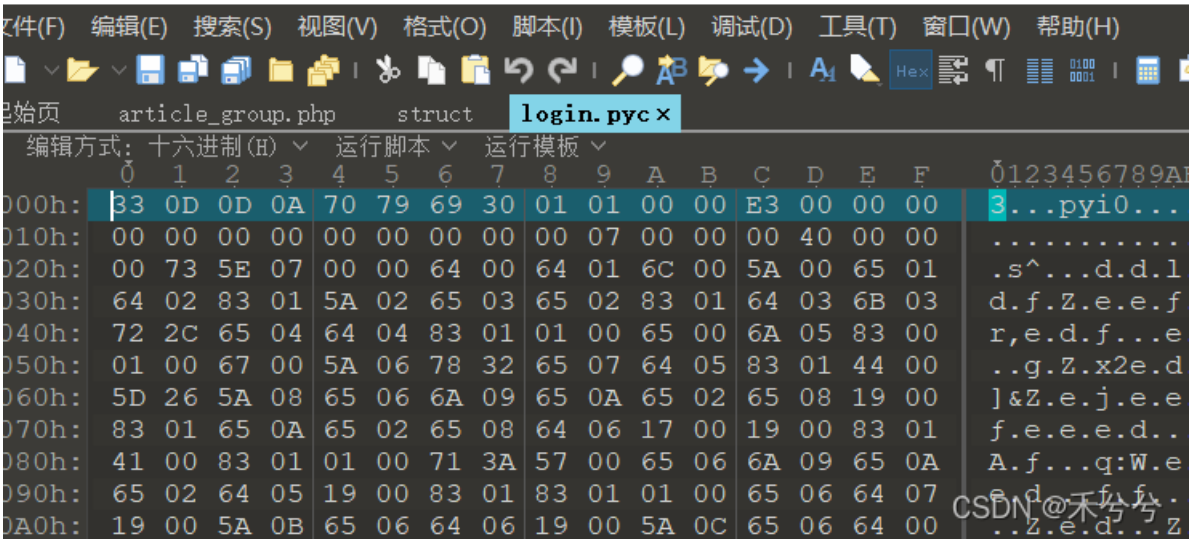
利用pyinstxtractor.py 进行反编译，修复生成文件下的login文件的头，我用到了010eidtor，修复方法参考了了这篇文章struct里面提供了各个pyc文件修复头

[分享]PyInstaller 解包-软件逆向-看雪论坛-安全社区|安全招聘|bbs.pediy.com




用010editor打开struct和login文件，将struct里面E3之前的字节复制到login里面




然后将login加上pyc后缀，利用uncompyle6将pyc文件反编译成py文件得的源代码：

```
# uncompyle6 version 3.8.0
# Python bytecode 3.6 (3379)
# Decompiled from: Python 3.8.10 (tags/v3.8.10:3d8993a, May  3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)]
# Embedded file name: login.py
# Compiled at: 1995-09-28 00:18:56
# Size of source mod 2**32: 257 bytes
import sys
input1 = input('input something:')
if len(input1) != 14:
    print('Wrong length!')
    sys.exit()
else:
    code = []
    for i in range(13):
        code.append(ord(input1[i]) ^ ord(input1[(i + 1)]))

    code.append(ord(input1[13]))
    a1 = code[2]
    a2 = code[1]
    a3 = code[0]
    a4 = code[3]
    a5 = code[4]
    a6 = code[5]
    a7 = code[6]
    a8 = code[7]
    a9 = code[9]
    a10 = code[8]
    a11 = code[10]
    a12 = code[11]
    a13 = code[12]
    a14 = code[13]
    if (a1 * 88 + a2 * 67 + a3 * 65 - a4 * 5 + a5 * 43 + a6 * 89 + a7 * 25 + a8 * 13 - a9 * 36 + a10 * 15 +
        print('flag is GWHT{md5(your_input)}')
        print('Congratulations and have fun!')
    else:
        print('Sorry,plz try again...')
# okay decompiling login.pyc
```

```
import hashlib
from z3 import *

a1=Int("a1")
a2=Int("a2")
a3=Int("a3")
a4=Int("a4")
a5=Int("a5")
a6=Int("a6")
a7=Int("a7")
a8=Int("a8")
a9=Int("a9")
a10=Int("a10")
a11=Int("a11")
a12=Int("a12")
a13=Int("a13")
a14=Int("a14")

a = Solver()

a.add(a1 * 88 + a2 * 67 + a3 * 65 - a4 * 5 + a5 * 43 + a6 * 89 + a7 * 25 + a8 * 13 - a9 * 36 + a10 * 15 + a
a.add(a1 * 89 + a2 * 7 + a3 * 12 - a4 * 25 + a5 * 41 + a6 * 23 + a7 * 20 - a8 * 66 + a9 * 31 + a10 * 8 + a1
a.add(a1 * 28 + a2 * 35 + a3 * 16 - a4 * 65 + a5 * 53 + a6 * 39 + a7 * 27 + a8 * 15 - a9 * 33 + a10 * 13 +
a.add(a1 * 23 + a2 * 34 + a3 * 35 - a4 * 59 + a5 * 49 + a6 * 81 + a7 * 25 + (a8 * 128) - a9 * 32 + a10 * 75
a.add(a1 * 38 + a2 * 97 + a3 * 35 - a4 * 52 + a5 * 42 + a6 * 79 + a7 * 90 + a8 * 23 - a9 * 36 + a10 * 57 +
a.add(a1 * 22 + a2 * 27 + a3 * 35 - a4 * 45 + a5 * 47 + a6 * 49 + a7 * 29 + a8 * 18 - a9 * 26 + a10 * 35 +
a.add(a1 * 12 + a2 * 45 + a3 * 35 - a4 * 9 - a5 * 42 + a6 * 86 + a7 * 23 + a8 * 85 - a9 * 47 + a10 * 34 + a
a.add(a1 * 79 + a2 * 62 + a3 * 35 - a4 * 85 + a5 * 33 + a6 * 79 + a7 * 86 + a8 * 14 - a9 * 30 + a10 * 25 +
a.add(a1 * 8 + a2 * 6 + a3 * 64 - a4 * 85 + a5 * 73 + a6 * 29 + a7 * 2 + a8 * 23 - a9 * 36 + a10 * 5 + a11
a.add(a1 * 67 - a2 * 68 + a3 * 68 - a4 * 51 - a5 * 43 + a6 * 81 + a7 * 22 - a8 * 12 - a9 * 38 + a10 * 75 +
a.add(a1 * 85 + a2 * 63 + a3 * 5 - a4 * 51 + a5 * 44 + a6 * 36 + a7 * 28 + a8 * 15 - a9 * 6 + a10 * 45 + a1
a.add(a1 * 47 + a2 * 64 + a3 * 66 - a4 * 5 + a5 * 43 + a6 * 112 + a7 * 25 + a8 * 13 - a9 * 35 + a10 * 95 +
a.add(a1 * 89 + a2 * 67 + a3 * 85 - a4 * 25 + a5 * 49 + a6 * 89 + a7 * 23 + a8 * 56 - a9 * 92 + a10 * 14 +
a.add(a1 * 95 + a2 * 34 + a3 * 62 - a4 * 9 - a5 * 43 + a6 * 83 + a7 * 25 + a8 * 12 - a9 * 36 + a10 * 16 + a

print(a.check())
print(a.model())
#注意login.py中a1 = code[2]，a3 = code[0]，a1和a3位置互换，a10和a9
a = [10, 24, 119, 7, 104, 43, 28, 91, 108, 52, 88, 74, 88, 33]
flag = ""

for i in range(12, -1, -1):
    a[i] = a[i] ^ a[i + 1]

print(a)
for i in a:
    flag += chr(i)
print(flag)
print(hashlib.md5(flag.encode(encoding='UTF-8')).hexdigest())
```

解密脚本如上