

buuctf——[羊城杯 2020]login

原创

re3sry 于 2021-09-30 18:08:05 发布 81 收藏

文章标签: [python buuctf reverse](#)

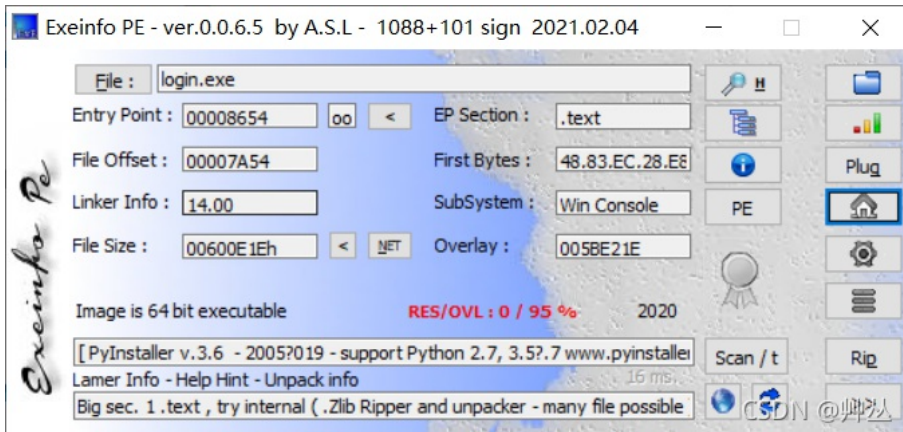
版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/yhfgs/article/details/120556096>

版权

1.查壳。

无壳, 64位。(当时还不知到PyInstaller)



2.直接丢到IDA反编译。发现啥也没有。

(连个提示性的字符串也没有, 但运行是有input something。很迷。)

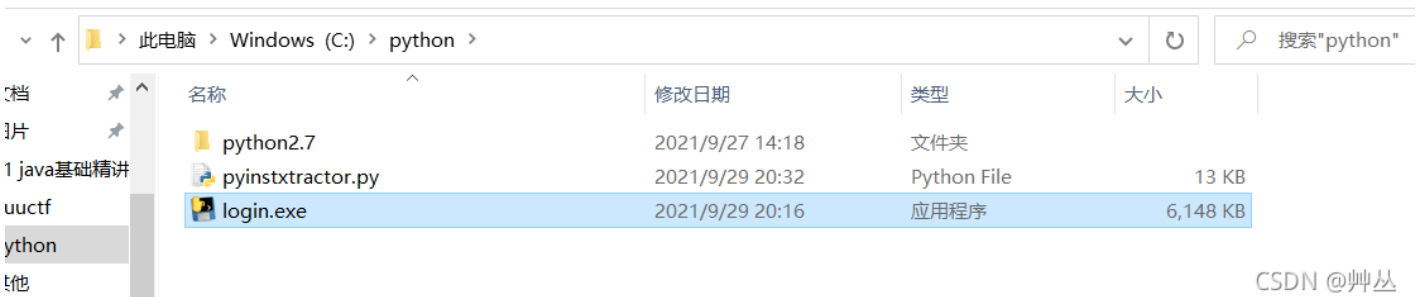
看了看大佬的wp, 才知道这是PyInstaller打包的exe, 需要解包。

3.解包, 反编译。

(1) 解包;

下载PyInstaller Extractor ([PyInstaller Extractor download | SourceForge.net](#))

把login.exe与下载得到的pyinstxtractor.py放到同一目录下。



cmd运行命令, 得到一个文件夹。

CSDN @ 帅丛

```

attempt to call a nil value
C:\其他\Cmder> cd C:\python

attempt to call a nil value
C:\python>pyinstxtractor.py login.exe
[*] Processing login.exe
[*] Pyinstaller version: 2.1+
[*] Python version: 36
[*] Length of package: 6021662 bytes
[*] Found 59 files in CArchive
[*] Beginning extraction...please standby
[+] Possible entry point: pyiboot01_bootstrap
[+] Possible entry point: login
[!] Warning: The script is running in a different python version than the one used to build the executable
Run this script in Python36 to prevent extraction errors(if any) during unmarshalling
[!] Unmarshalling FAILED. Cannot extract PYZ-00.pyz. Extracting remaining files.
[*] Successfully extracted pyinstaller archive: login.exe

You can now use a python decompiler on the pyc files within the extracted directory

attempt to call a nil value
C:\python>

```

CSDN @艸丛

login.exe_extracted	2021/9/29 20:53	文件夹	
python2.7	2021/9/27 14:18	文件夹	
login.exe	2021/9/29 20:16	应用程序	6,148 KB
pyinstxtractor.py	2021/9/29 20:32	Python File	13 KB

CSDN @艸丛

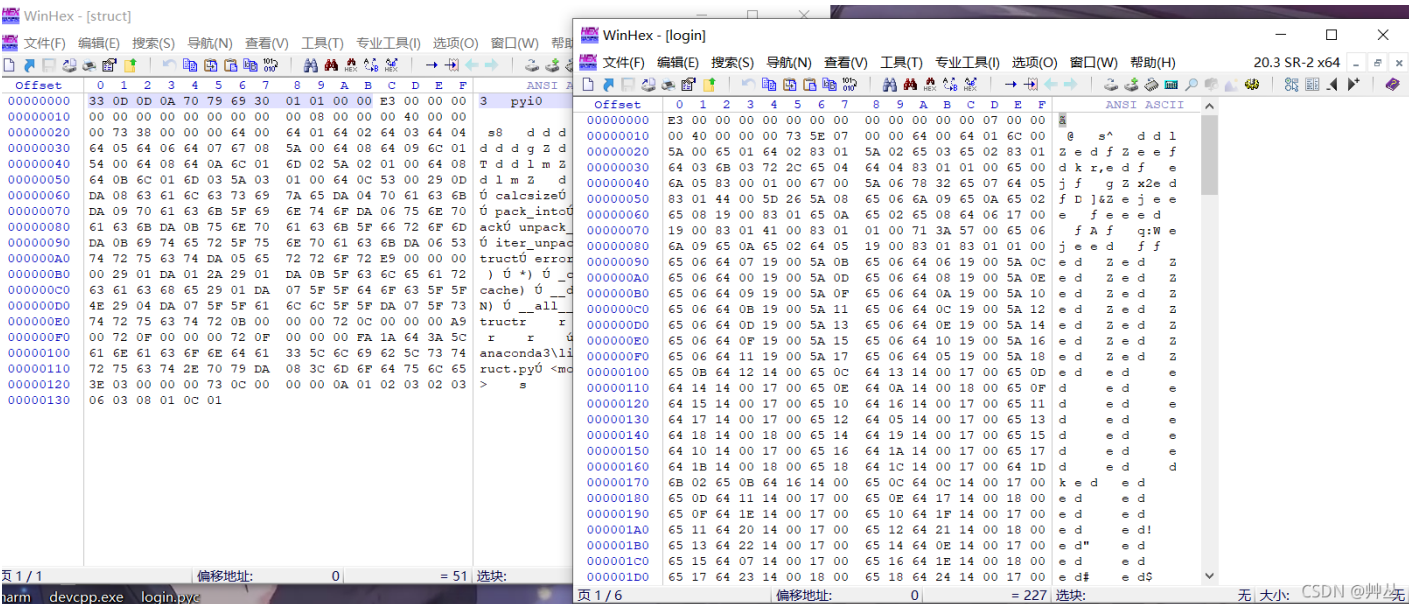
在文件夹中，找到login和struct（无后缀名。）

用winhex把struct和login打开。

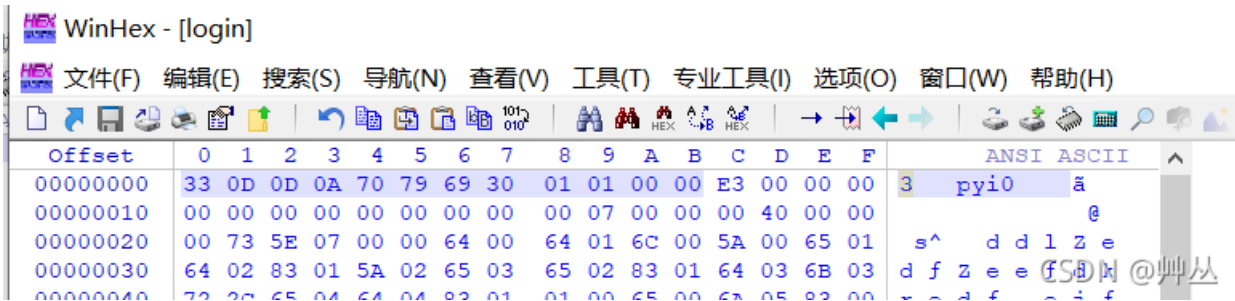
把login前面缺的字节补上（在struct中），

login这个是在从E3开始的，所以要把struct中E3前的字节都复制过去。

开始时：



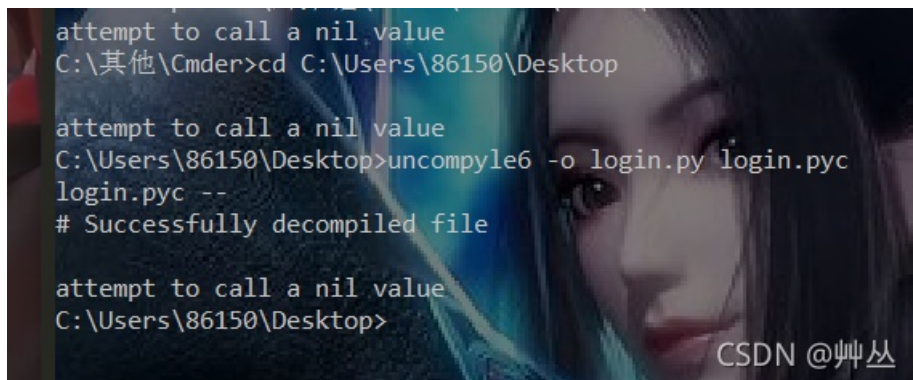
复制后：



并以.pyc为后缀名保存。

(2) 用uncompyle反编译。(uncompyle6安装使用方法 - pcat - 博客园)

得到.py文件。



4. 打开py文件进行分析。

```
login.py - C:\Users\86150\Desktop\login.py (2.7.18)
File Edit Format Run Options Window Help
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
    & (a1 * 89 + a2 * 7 + a3 * 12 - a4 * 25 + a5 * 41 + a6 * 23 + a7 * 20 - a8 *
    (a1 * 28 + a2 * 35 + a3 * 16 - a4 * 65 + a5 * 53 + a6 * 39 + a7 * 27 + a8 *
    (a1 * 23 + a2 * 34 + a3 * 35 - a4 * 59 + a5 * 49 + a6 * 81 + a7 * 25 + a8 *
    (a1 * 38 + a2 * 97 + a3 * 35 - a4 * 52 + a5 * 42 + a6 * 79 + a7 * 90 + a8 *
    (a1 * 22 + a2 * 27 + a3 * 35 - a4 * 45 + a5 * 47 + a6 * 49 + a7 * 29 + a8 *
    (a1 * 12 + a2 * 45 + a3 * 35 - a4 * 9 - a5 * 42 + a6 * 86 + a7 * 23 + a8 * 8
    (a1 * 79 + a2 * 62 + a3 * 35 - a4 * 85 + a5 * 33 + a6 * 79 + a7 * 86 + a8 *
    (a1 * 8 + a2 * 6 + a3 * 64 - a4 * 85 + a5 * 73 + a6 * 29 + a7 * 2 + a8 * 23
    (a1 * 67 - a2 * 68 + a3 * 68 - a4 * 51 - a5 * 43 + a6 * 81 + a7 * 22 - a8 *
    (a1 * 85 + a2 * 63 + a3 * 5 - a4 * 51 + a5 * 44 + a6 * 36 + a7 * 28 + a8 * 1
    (a1 * 47 + a2 * 64 + a3 * 66 - a4 * 5 + a5 * 43 + a6 * 112 + a7 * 25 + a8 *
    (a1 * 89 + a2 * 67 + a3 * 85 - a4 * 25 + a5 * 49 + a6 * 89 + a7 * 23 + a8 *
    (a1 * 95 + a2 * 34 + a3 * 62 - a4 * 9 - a5 * 43 + a6 * 83 + a7 * 25 + a8 * 1
    print('flag is GWHT{md5(your_input)}')
    print('Congratulations and have fun!')
```

一个简单异或加解方程（z3）最后MD5加密。

上脚本：

```

import hashlib
#解方程
from z3 import*

a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12, a13, a14 = Ints('a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14')

x=Solver()
x.add(a1 * 88 + a2 * 67 + a3 * 65 - a4 * 5 + a5 * 43 + a6 * 89 + a7 * 25 + a8 * 13 - a9 * 36 + a10 * 15 + a11 * 11 + a12 * 47 - a13 * 60 + a14 * 29 == 22748)
x.add(a1 * 89 + a2 * 7 + a3 * 12 - a4 * 25 + a5 * 41 + a6 * 23 + a7 * 20 - a8 * 66 + a9 * 31 + a10 * 8 + a11 * 2 - a12 * 41 - a13 * 39 + a14 * 17 == 7258)
x.add(a1 * 28 + a2 * 35 + a3 * 16 - a4 * 65 + a5 * 53 + a6 * 39 + a7 * 27 + a8 * 15 - a9 * 33 + a10 * 13 + a11 * 101 + a12 * 90 - a13 * 34 + a14 * 23 == 26190)
x.add(a1 * 23 + a2 * 34 + a3 * 35 - a4 * 59 + a5 * 49 + a6 * 81 + a7 * 25 + a8 * 128 - a9 * 32 + a10 * 75 + a11 * 81 + a12 * 47 - a13 * 60 + a14 * 29 == 37136)
x.add(a1 * 38 + a2 * 97 + a3 * 35 - a4 * 52 + a5 * 42 + a6 * 79 + a7 * 90 + a8 * 23 - a9 * 36 + a10 * 57 + a11 * 81 + a12 * 42 - a13 * 62 - a14 * 11 == 27915)
x.add(a1 * 22 + a2 * 27 + a3 * 35 - a4 * 45 + a5 * 47 + a6 * 49 + a7 * 29 + a8 * 18 - a9 * 26 + a10 * 35 + a11 * 41 + a12 * 40 - a13 * 61 + a14 * 28 == 17298)
x.add(a1 * 12 + a2 * 45 + a3 * 35 - a4 * 9 - a5 * 42 + a6 * 86 + a7 * 23 + a8 * 85 - a9 * 47 + a10 * 34 + a11 * 76 + a12 * 43 - a13 * 44 + a14 * 65 == 19875)
x.add(a1 * 79 + a2 * 62 + a3 * 35 - a4 * 85 + a5 * 33 + a6 * 79 + a7 * 86 + a8 * 14 - a9 * 30 + a10 * 25 + a11 * 11 + a12 * 57 - a13 * 50 - a14 * 9 == 22784)
x.add(a1 * 8 + a2 * 6 + a3 * 64 - a4 * 85 + a5 * 73 + a6 * 29 + a7 * 2 + a8 * 23 - a9 * 36 + a10 * 5 + a11 * 2 + a12 * 47 - a13 * 64 + a14 * 27 == 9710)
x.add(a1 * 67 - a2 * 68 + a3 * 68 - a4 * 51 - a5 * 43 + a6 * 81 + a7 * 22 - a8 * 12 - a9 * 38 + a10 * 75 + a11 * 41 + a12 * 27 - a13 * 52 + a14 * 31 == 13376)
x.add(a1 * 85 + a2 * 63 + a3 * 5 - a4 * 51 + a5 * 44 + a6 * 36 + a7 * 28 + a8 * 15 - a9 * 6 + a10 * 45 + a11 * 31 + a12 * 7 - a13 * 67 + a14 * 78 == 24065)
x.add(a1 * 47 + a2 * 64 + a3 * 66 - a4 * 5 + a5 * 43 + a6 * 112 + a7 * 25 + a8 * 13 - a9 * 35 + a10 * 95 + a11 * 21 + a12 * 43 - a13 * 61 + a14 * 20 == 27687)
x.add(a1 * 89 + a2 * 67 + a3 * 85 - a4 * 25 + a5 * 49 + a6 * 89 + a7 * 23 + a8 * 56 - a9 * 92 + a10 * 14 + a11 * 89 + a12 * 47 - a13 * 61 - a14 * 29 == 29250)
x.add(a1 * 95 + a2 * 34 + a3 * 62 - a4 * 9 - a5 * 43 + a6 * 83 + a7 * 25 + a8 * 12 - a9 * 36 + a10 * 16 + a11 * 51 + a12 * 47 - a13 * 60 - a14 * 24 == 15317)

check = x.check()

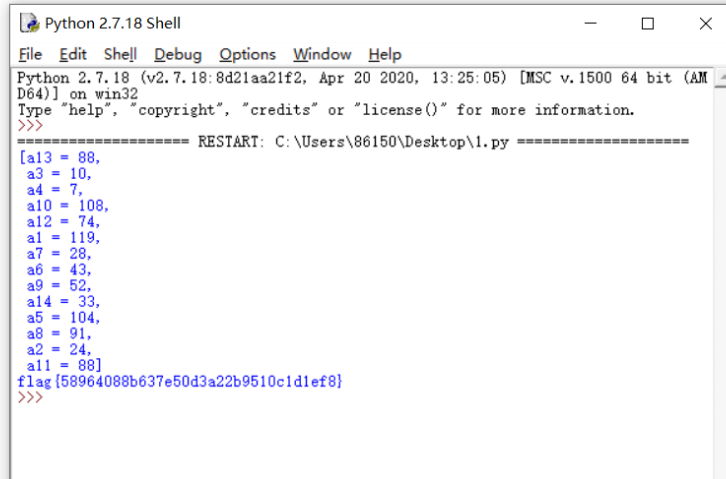
print(x.model())

#异或
model=[119, 24, 10, 7, 104, 43, 28, 91, 52, 108, 88, 74, 88, 33]
new=[10, 24, 119, 7, 104, 43, 28, 91, 108, 52, 88, 74, 88, 33]
flag=""

for i in range(12, -1, -1):
    new[i] = new[i] ^ new[i+1]
for i in range(len(new)):
    flag += chr(new[i])

#MD5加密
m=hashlib.md5()
m.update(flag.encode('utf-8'))
print('flag{' + m.hexdigest() + '}')

```



CSDN @帅丛

5.get flag

flag{58964088b637e50d3a22b9510c1d1ef8}