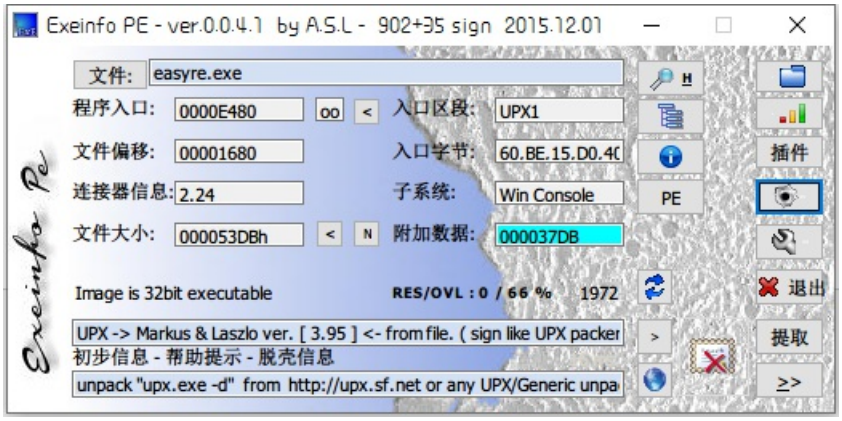
BUUCTF--[ACTF新生赛2020]easyre

原创

Hk_Mayfly 于 2020-04-09 01:46:00 发布 1629 收藏 1

测试文件：**https://www.lanzous.com/ib515vi**

脱壳



获取到信息

- 32位文件
- upx加密

```
C:\Users\10245\Desktop>upx -d easyre.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2018
UPX 3.95w       Markus Oberhumer, Laszlo Molnar & John Reiser   Aug 26th 2018

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     28123 <-     21467   76.33%    win32/pe     easyre.exe

Unpacked 1 file.
```

代码分析

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char v4; // [esp+12h] [ebp-2Eh]
  char v5; // [esp+13h] [ebp-2Dh]
  char v6; // [esp+14h] [ebp-2Ch]
  char v7; // [esp+15h] [ebp-2Bh]
  char v8; // [esp+16h] [ebp-2Ah]
  char v9; // [esp+17h] [ebp-29h]
  char v10; // [esp+18h] [ebp-28h]
  char v11; // [esp+19h] [ebp-27h]
  char v12; // [esp+1Ah] [ebp-26h]
  char v13; // [esp+1Bh] [ebp-25h]
  char v14; // [esp+1Ch] [ebp-24h]
  char v15; // [esp+1Dh] [ebp-23h]
  int v16; // [esp+1Eh] [ebp-22h]
  int v17; // [esp+22h] [ebp-1Eh]
  int v18; // [esp+26h] [ebp-1Ah]
  __int16 v19; // [esp+2Ah] [ebp-16h]
  char v20; // [esp+2Ch] [ebp-14h]
  char v21; // [esp+2Dh] [ebp-13h]
  char v22; // [esp+2Eh] [ebp-12h]
  int v23; // [esp+2Fh] [ebp-11h]
  int v24; // [esp+33h] [ebp-Dh]
  int v25; // [esp+37h] [ebp-9h]
  char v26; // [esp+3Bh] [ebp-5h]
  int i; // [esp+3Ch] [ebp-4h]

  __main();
  v4 = 42;
  v5 = 70;
  v6 = 39;
  v7 = 34;
  v8 = 78;
  v9 = 44;
  v10 = 34;
  v11 = 40;
  v12 = 73;
  v13 = 63;
  v14 = 43;
  v15 = 64;
  printf("Please input:");
  scanf("%s", &v19);
  if ( (_BYTE)v19 != 65 || HIBYTE(v19) != 67 || v20 != 84 || v21 != 70 || v22 != 123 || v26 != 125 )
    return 0;
  v16 = v23;
  v17 = v24;
  v18 = v25;
  for ( i = 0; i <= 11; ++i )
  {
    if ( *(&v4 + i) != _data_start__[*((char *)&v16 + i) - 1] )
      return 0;
  }
  printf("You are correct!");
  return 0;
}
```

着眼观察for循环就行，从for循环了解到flag长度应该是12，将flag的ASCII值作为下标取值，与v4数组比较。很简单，只需要利用v4数组在_data_start__中找位置，就是我们flag的值

```
.data:00402000 ; char _data_start__[]
.data:00402000 __data_start__  db 7Eh                 ; DATA XREF: _main+ECtr
.data:00402001 aZyxwvutsrqponm db '}|{zyxwvutsrqponmlkjihgfedcba`_^]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>='
.data:00402001                 db '<;:9876543210/.-,+*)(',27h,'&%$# !"',0
.data:00402060                 align 40h
.data:00402080                 public __CRT_glob
.data:00402080 __CRT_glob      dd 0FFFFFFFFh          ; DATA XREF: ___mingw_CRTStartup+4Atr
```

脚本

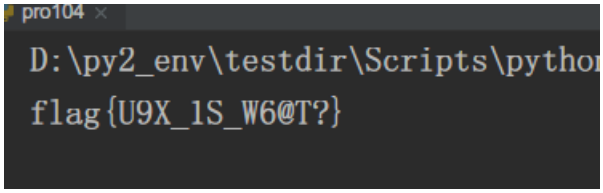
```
# -*- coding:utf-8 -*-

v4 = [42,70,39,34,78,44,34,40,73,63,43,64]

model = r"}|{zyxwvutsrqponmlkjihgfedcba`_^]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>=<;:9876543210/.-,+*)(" + chr(0x2

pos = []

for i in v4:
    pos.append(model.find(chr(i))+1)
s = [chr(x + 1) for x in pos]
flag = ''.join(s)
print ('flag{'+flag+'}')
```



get flag！

> **flag{U9X_1S_W6@T?}**