buu-[ACTF新生赛2020]easyre

原创

有点水啊 于 2022-02-08 01:35:57 发布 1734 收藏

分类专栏： buuctf-reserve 文章标签： 安全

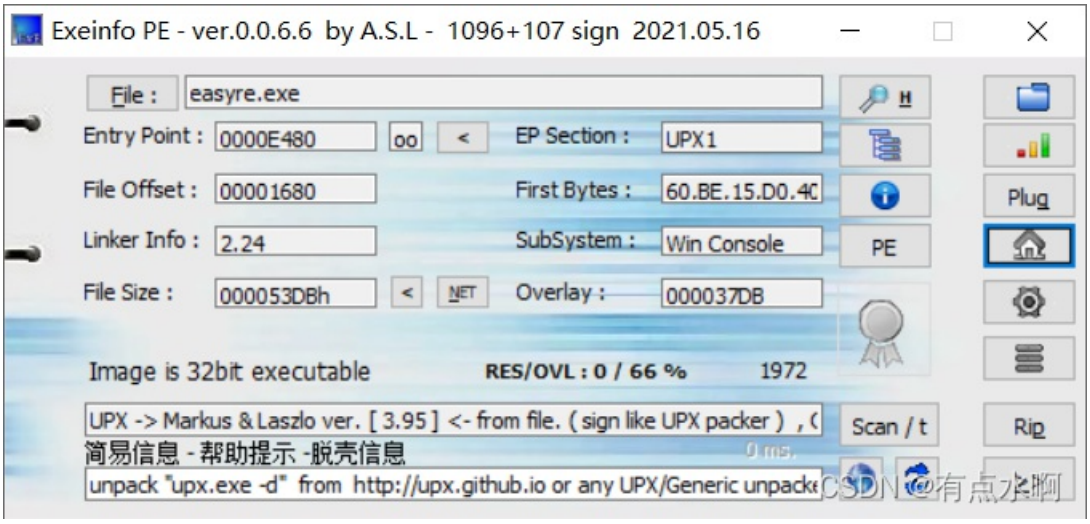
本文链接：https://blog.csdn.net/qaq517384/article/details/122816667

buuctf-reserve 专栏收录该内容

59 篇文章 0 订阅

订阅专栏

查看文件信息



32位，UPX脱壳

查看main函数

```
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]

  sub_401A10();
  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 ( v19 != 'A' || HIBYTE(v19) != 'C' || v20 != 'T' || v21 != 'F' || v22 != '{' || v26 != '}' )
 //ACTF{}
    return 0;
  v16 = v23;
  v17 = v24;
  v18 = v25;
  for ( i = 0; i <= 11; ++i )
  {
    if ( *(&v4 + i) != byte_402000[*((char *)&v16 + i) - 1] )
      return 0;
  }
  printf("You are correct!");
  return 0;
}
```

byte_402000,包括7E的

```
UPX0:00402000 ; char byte_402000[]
UPX0:00402000 byte_402000     db 7Eh                  ; DATA XREF: _main+EC↑r
UPX0:00402001 aZyxwvutsrqponm db '}|{zyxwvutsrqponmlkjihgfedcba`_^]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>='
UPX0:00402001                 db '<;:9876543210/.-,+*)(',27h,'&%$# !"',0
UPX0:00402060                 align 40h
```

16进制视角

```
00402000  7E 7D 7C 7B 7A 79 78 77  76 75 74 73 72 71 70 6F  ~}|{zyxwvutsrqpo
00402010  6E 6D 6C 6B 6A 69 68 67  66 65 64 63 62 61 60 5F  nmlkjihgfedcba`_
00402020  5E 5D 5C 5B 5A 59 58 57  56 55 54 53 52 51 50 4F  ^]\[ZYXWVUTSRQPO
00402030  4E 4D 4C 4B 4A 49 48 47  46 45 44 43 42 41 40 3F  NMLKJIHGFEDCBA@?
00402040  3E 3D 3C 3B 3A 39 38 37  36 35 34 33 32 31 30 2F  >=<;:9876543210/
00402050  2E 2D 2C 2B 2A 29 28 27  26 25 24 23 20 21 22 00  .-,+*)('&%$#·!".
00402060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

for循环0-11应该就是12位的flag
数组v4的值是byte_402000以[v16+i-1]为下标所对应的值

放上exp

```python
byte_402000 = '~}|{zyxwvutsrqponmlkjihgfedcba`_^]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>=<;:9876543210/.-,+*)(\'&%$# !"'

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

flag = ''

for i in v4:
    flag += chr(byte_402000.find(chr(i)) + 1)

print(flag)
```

`flag{U9X_1S_W6@T?}`

对flag存放的地址没太搞懂
翻了好几页的wp发现有些wp的逆向是这样的，可能是脱壳工具不一样

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  _BYTE v4[12]; // [esp+12h] [ebp-2Eh] BYREF
  _DWORD v5[3]; // [esp+1Eh] [ebp-22h]
  _BYTE v6[5]; // [esp+2Ah] [ebp-16h] BYREF
  int v7; // [esp+2Fh] [ebp-11h]
  int v8; // [esp+33h] [ebp-Dh]
  int v9; // [esp+37h] [ebp-9h]
  char v10; // [esp+3Bh] [ebp-5h]
  int i; // [esp+3Ch] [ebp-4h]

  sub_401A10();
  qmemcpy(v4, "*F'\"N,\"(I?+@", sizeof(v4));
  printf("Please input:");
  scanf("%s", v6);
  if ( v6[0] != 'A' || v6[1] != 'C' || v6[2] != 'T' || v6[3] != 'F' || v6[4] != '{' || v10 != '}' )
    return 0;
  v5[0] = v7;
  v5[1] = v8;
  v5[2] = v9;
  for ( i = 0; i <= 11; ++i )
  {
    if ( v4[i] != byte_402000[*((char *)v5 + i) - 1] )
      return 0;
  }
  printf("You are correct!");
  return 0;
}
```

v4是*F'“N,”(I?+@
然后flag存在v5里
v5分为三组，v7,v8,v9
（恍然大悟.jpg）
漏了这一步，三个int刚好12字节，对应v4-v15
v19-v22对应ACTF{
v23-v25为flag
v26为}

```
v16 = v23;
v17 = v24;
v18 = v25;
for ( i = 0;
```

```
int v23; // [esp+2Fh] [ebp-11h]
int v24; // [esp+33h] [ebp-Dh]
int v25; // [esp+37h] [ebp-9h]
```

翻了7页的百度，下次要注意了