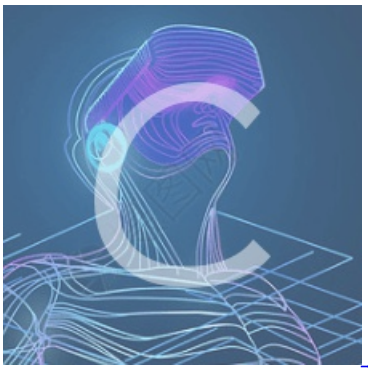
BUUCTF_[ACTF新生赛2020]easyre

原创

ZYen12138 于 2020-11-05 22:48:36 发布

分类专栏： # BUUCTF CTF

本文链接：https://blog.csdn.net/weixin_46009088/article/details/109521777



BUUCTF 同时被 2 个专栏收录

17 篇文章 2 订阅

订阅专栏

CTF

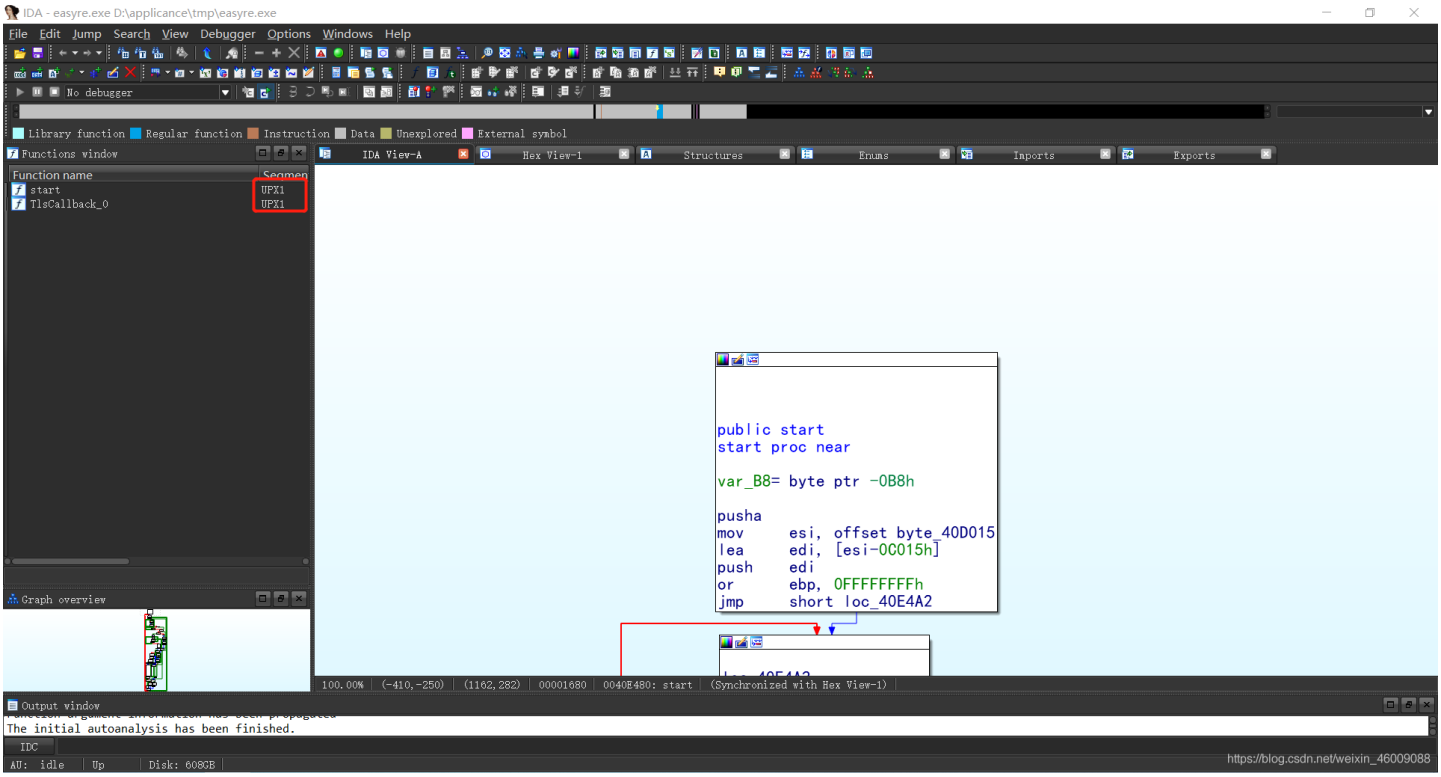
17 篇文章 0 订阅

订阅专栏

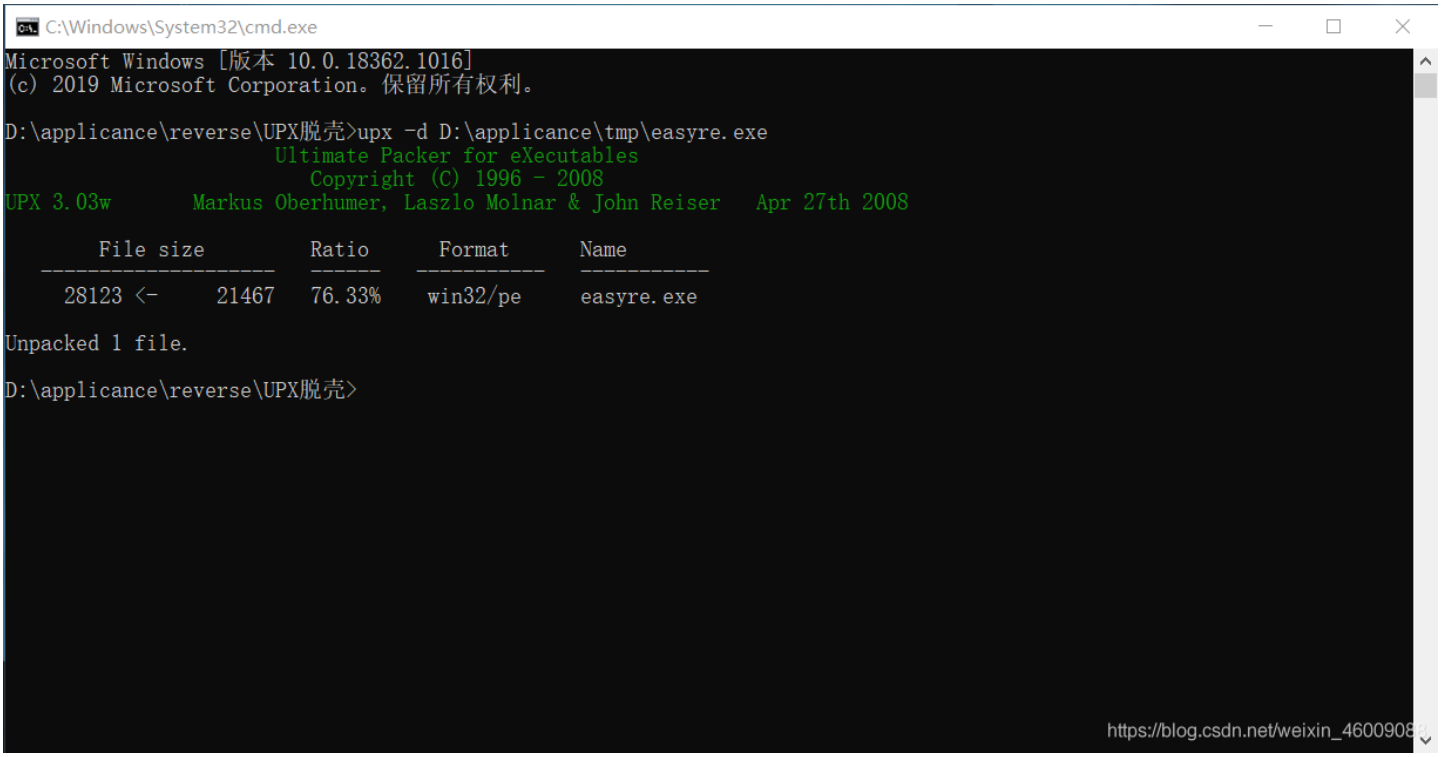
BUUCTF_[ACTF新生赛2020]easyre

给IDA坑了，学到了学到了！

用IDA打开，有壳：



是exe文件，直接脱壳：



找到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]

  __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;
}
```

关键语句：

```
for ( i = 0; i <= 11; ++i )
{
  if ( *(&v4 + i) != _data_start__[*((char *)&v16 + i) - 1] )
    return 0;
}
```

点进_data_start__ 查看字符串，一点要注意上面的7Eh也是算进去的.

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

打开Hex View看会比较好一点：

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

接下来就可以写python脚本了，如图：

```python
key = '~}|{zyxwvutsrqponmlkjihgfedcba`_^]\[ZYXWVUTSRQPONMLKJIHGFEDCBA@?>=<;:9876543210/.-,+*)(\'&%$# !"' #'一定要加\
encrypt = [42,70,39,34,78,44,34,40,73,63,43,64]
x = []
flag = ''
for i in encrypt:
  x.append(key.find(chr(i))+1)
for i in x:
  flag += chr(i)
print(flag)
```

总结：

①*(&v4 + i) !=*data_start*_[*((char *)&v16 + i) - 1]) 的意思是在_data_start__字符串里面寻找一个字符然后 -1 与v4进行对比，注意括号的位置就能知道 -1 是减到索引还是索引值上，所以逆过来就是 +1 .

②IDA的字符串要进Hex View查看.