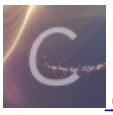
攻防世界Reverse进阶区-IgniteMe-writeup

原创

y4ung 于 2020-09-19 09:30:28 发布 300 收藏 1

分类专栏： ctf 文章标签： ctf

本文链接：https://blog.csdn.net/qq_35056292/article/details/108676873



1. 介绍

本题是xctf攻防世界中Reverse的进阶区的题IgniteMe

题目来源： 高校网络信息安全运维挑战赛

2. 分析

```
$ file fac4d1290e604fdfacbbe06fd1a5ca39.exe
fac4d1290e604fdfacbbe06fd1a5ca39.exe: PE32 executable (console) Intel 80386, for MS Windows
```

```
C:\Users\hzy\Downloads>fac4d1290e604fdfacbbe06fd1a5ca39.exe
Give me your flag:
123
Sorry, keep trying!
```

扔进IDA里， `shift+F12` 查看strings window。找到字符串 `Give me your flag:` 被引用的地方在函数main里头。F5查看伪代码。

很明显可以看出，用户输入的长度必须大于4且小于30，以 `EIS{` 开头，以 `}` 结尾。并且，在sub_4011C0函数中，对用户的输入进行了校验

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int result; // eax
  size_t i; // [esp+4Ch] [ebp-8Ch]
  char v5[4]; // [esp+50h] [ebp-88h]
  char v6[28]; // [esp+58h] [ebp-80h]
  char v7; // [esp+74h] [ebp-64h]

  printf(&unk_446360, "Give me your flag:");
  sub_4013F0(sub_403670);
  sub_401440(v6, 127);
  if ( strlen(v6) < 30 && strlen(v6) > 4 )
  {
    strcpy(v5, "EIS{");                          // 以EIS{开头
    for ( i = 0; i < strlen(v5); ++i )
    {
      if ( v6[i] != v5[i] )
      {
        printf(&unk_446360, "Sorry, keep trying! ");
        sub_4013F0(sub_403670);
        return 0;
      }
    }
    if ( v7 == '}' )                             // 以}结尾
    {
      if ( sub_4011C0(v6) )
        printf(&unk_446360, "Congratulations! ");
      else
        printf(&unk_446360, "Sorry, keep trying! ");
      sub_4013F0(sub_403670);
      result = 0;
    }
    else
    {
      printf(&unk_446360, "Sorry, keep trying! ");
      sub_4013F0(sub_403670);
      result = 0;
    }
  }
  else
  {
    printf(&unk_446360, "Sorry, keep trying!");
    sub_4013F0(sub_403670);
    result = 0;
  }
  return result;
}
```


跟进sub_4011C0函数中。首先把用户输入从下标为[4]开始的字符（不包括最后一个字符）赋值给数组v8，v4全部用0填充，这里的v4，从最后一行来看，就是最后从用户输入变换后得到的字符串，并且用户输入中在EIS{}中间的字符数应该为24个字符。接下来对于v8中的每个字符v8[i]：

- 如果v8[i]为小写，则转成大写；
- 如果v8[i]为大写，则转成小写；

```
bool __cdecl sub_4011C0(char *user_input)
{
  size_t v2; // eax
  signed int v3; // [esp+50h] [ebp-B0h]
  char v4[32]; // [esp+54h] [ebp-ACh]
  int v5; // [esp+74h] [ebp-8Ch]
  int v6; // [esp+78h] [ebp-88h]
  size_t i; // [esp+7Ch] [ebp-84h]
  char v8[128]; // [esp+80h] [ebp-80h]

  if ( strlen(user_input) <= 4 )
    return 0;
  i = 4;
  v6 = 0;
  while ( i < strlen(user_input) - 1 )
    v8[v6++] = user_input[i++];                 // 把user_input [4]以后的字符赋值给字符串/数组 v8
  v8[v6] = 0;                                   // v8最后一位赋值为0
  v5 = 0;
  v3 = 0;
  memset(v4, 0, 32u);                           // v4的32个字节都用0填充
  for ( i = 0; ; ++i )
  {
    v2 = strlen(v8);
    if ( i >= v2 )
      break;
    if ( v8[i] >= 'a' && v8[i] <= 'z' )
    {
      v8[i] -= 32;                              // 小写字母转大写字母
      v3 = 1;
    }
    if ( !v3 && v8[i] >= 65 && v8[i] <= 90 )    // 大写转小写
      v8[i] += 32;
    v4[i] = byte_4420B0[i] ^ sub_4013C0(v8[i]);
    v3 = 0;
  }
  return strcmp("GONDPHyGjPEKruv{{pj]X@rF", v4) == 0;
}
```


然后`v4[i] = byte_4420B0[i] ^ sub_4013C0(v8[i]); = byte_4420B0[i] ^ ( (v8[i] ^ 85) + 72 )`。

```
int __cdecl sub_4013C0(int v8[i])
{
    return (v8[i] ^ 85) + 72;
}
```

其中， byte_4420B0的内容为：`0D 13 17 11 02 01 20 1D 0C 02 19 2F 17 2B 24 1F 1E 16 09 0F 15 27 13 26`

编写python脚本逆出原字符串，得到flag：`EIS{wadx_tdgk_aihc_ihkn_pjlm}`

```
byte_4420B0 = "0D 13 17 11 02 01 20 1D 0C 02 19 2F 17 2B 24 1F 1E 16 09 0F 15 27 13 26"
byte_4420B0_split = byte_4420B0.split(" ")
target = "GONDPHyGjPEKruv{{pj]X@rF"
res = ""  # ESI{} 中的部分
for i, each in enumerate(target):
    sub_4013C0_value = ord(each) ^ int(byte_4420B0_split[i], 16)
    v8_i = (sub_4013C0_value - 72) ^ 85
    v8_i = chr(v8_i)
    v3 = False
    if v8_i.isupper():
        v8_i = v8_i.lower()
        v3 = True
    if not v3 and v8_i.islower():
        v8_i = v8_i.upper()

    res += v8_i

print("flag: EIS{{{0}}}".format(res))
```