BUUCTF Reverse SimpleRev WriteUp

原创

PlumpBoy 于 2021-09-10 18:44:08 发布 578 收藏

分类专栏： BUUCTF 逆向题解 文章标签： c语言 系统安全 安全

simplerev-WP

使用exeinfope查看发现没壳，扔入IDA

```
int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
{
  int v3; // eax
  char v4; // [rsp+Fh] [rbp-1h]

  while ( 1 )
  {
    while ( 1 )
    {
      printf("Welcome to CTF game!\nPlease input d/D to start or input q/Q to quit this program: ");
      v4 = getchar();
      if ( v4 != 100 && v4 != 68 )
        break;
      Decry();
    }
    if ( v4 == 113 || v4 == 81 )
      Exit("Welcome to CTF game!\nPlease input d/D to start or input q/Q to quit this program: ", argv);
    puts("Input fault format!");
    v3 = getchar();
    putchar(v3);
  }
}
```

从主函数中可以看出，核心部分是Decry()函数，接下来对其进行分析。

```
unsigned __int64 Decry()
{
    char v1; // [rsp+Fh] [rbp-51h]
    int v2; // [rsp+10h] [rbp-50h]
    int v3; // [rsp+14h] [rbp-4Ch]
    int i; // [rsp+18h] [rbp-48h]
    int v5; // [rsp+1Ch] [rbp-44h]
    char src[8]; // [rsp+20h] [rbp-40h] BYREF
    __int64 v7; // [rsp+28h] [rbp-38h]
    int v8; // [rsp+30h] [rbp-30h]
    __int64 v9[2]; // [rsp+40h] [rbp-20h] BYREF
```

```
int v10; // [rsp+50h] [rbp-10h]
unsigned __int64 v11; // [rsp+58h] [rbp-8h]

char str2[104];

v11 = __readfsqword(0x28u);
*(_QWORD*)src = 0x534C43444ELL;//直接转换出来的结果是SLCDN
v7 = 0LL;
v8 = 0;
v9[0] = 0x776F646168LL;//直接转换出来的结果是wodah
v9[1] = '\0';
v10 = 0;
text = join(key3, (const char*)v9);//text=killshadow
strcpy(key, key1);
strcat(key, src);//key=ADSFKNDCLS
v2 = 0;
v3 = 0;
getchar();
v5 = strlen(key);//v5=10
for (i = 0; i < v5; ++i)//把大写字母转换为小写字母
{
    if (key[v3 % v5] > '@' && key[v3 % v5] <= 'Z')
        key[i] = key[v3 % v5] + 32;
    ++v3;
}

//key=adsfkndcls,v3=10
printf("Please input your flag:");
while (1)
{
    v1 = getchar();
    if (v1 == '\n')
        break;
    if (v1 == ' ')
    {
        ++v2;//v2代表str2的下标
    }
    else
    {
        if (v1 <= 96 || v1 > 122)
        {
            if (v1 > 64 && v1 <= 90)//大写字母
            {
                str2[v2] = (v1 - 39 - key[v3 % v5] + 97) % 26 + 97;
                ++v3;
            }
        }
        else//小写字母的处理
        {
            str2[v2] = (v1 - 39 - key[v3 % v5] + 97) % 26 + 97;
            ++v3;
        }
        if (!(v3 % v5))
            putchar(32);
        ++v2;
    }
}
if (!strcmp(text, str2))
    puts("Congratulation!\n");
else
```

```
        puts("Try again!\n");
    return __readfsqword(0x28u) ^ v11;
}
```

特别注意其中，用IDA的R键直接转换出来的有问题，因为是按照小端序排列的，但是直接转换时没有按照小端序进行修正，导致出来的是头尾颠倒的，注意，小端序的是2个十六进制（1字节）为一个单位，所以建议先转换，在调转。

写出解密脚本，这个是网上找的，当时自己直接手算的

```
#include <iostream>

using namespace std;

int main()
{
 char key[] = "adsfkndcls";
 char text[] = "killshadow";
 int v3 = 10;
 for (int i = 0; i < 10; i++)
 {
  for (int j = 0; j < 128; j++)
  {
   if (j < 'A' || j > 'z' || j > 'Z' && j < 'a')
   {
    continue;
   }
   if ((j - 39 - key[v3 % 10] + 97) % 26 + 97 == text[i])
   {
    cout << char(j);
    v3++;
    break;
   }
  }
 }
}
```

得到flag `flag{KLDQCUDFZO}`