buuctf 之 simplerev

原创

不会掉发的小鲁 于 2020-07-20 13:07:52 发布 1355 收藏 1

分类专栏： ctf 安全

本文链接：https://blog.csdn.net/weixin_47158947/article/details/107461010


ctf 同时被 2 个专栏收录

16 篇文章 0 订阅

订阅专栏


安全

15 篇文章 0 订阅

订阅专栏

1.这个为64位文件

需要学到的知识：小端序存储

拖入ida打开

```
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: ", argv
      v4 = getchar();
      if ( v4 != 100 && v4 != 68 )
        break;
      Decry();
    }
    if ( v4 == 113 || v4 == 81 )
      Exit();
    puts("Input fault format!");
    v3 = getchar();
    putchar(v3);
  }
}
```


通过分析可知，最重要的部分是在Decry（）函数

打开dercy函数

```
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]
__int64 v7; // [rsp+28h] [rbp-38h]
int v8; // [rsp+30h] [rbp-30h]
__int64 v9; // [rsp+40h] [rbp-20h]
__int64 v10; // [rsp+48h] [rbp-18h]
int v11; // [rsp+50h] [rbp-10h]
unsigned __int64 v12; // [rsp+58h] [rbp-8h]

v12 = __readfsqword(0x28u);
*(_QWORD *)src = 'SLCDN';
v7 = '\0';
v8 = 0;
v9 = 'wodah';
v10 = 0LL;
v11 = 0;
text = join(key3, (const char *)&v9);//就是这小端序存储 text=killshadow
strcpy(key, key1);//key="ADSFK"
strcat(key, src);//key="ADSFKNDCLS"小端序存储
v2 = 0;
v3 = 0;
getchar();
v5 = strlen(key);
for ( i = 0; i < v5; ++i )
{
  if ( key[v3 % v5] > 64 && key[v3 % v5] <= 90 )
    key[i] = key[v3 % v5] + 32;
  ++v3;
}//通过分析就可以知道，这是大写变小写
printf("Please input your flag:", src);
while ( 1 )
{
  v1 = getchar();
  if ( v1 == 10 )
    break;
  if ( v1 == 32 )
  {
    ++v2;
  }
  else
  {
    if ( v1 <= 96 || v1 > 122 )
    {
      if ( v1 > 64 && v1 <= 90 )
        str2[v2] = (v1 - 39 - key[v3++ % v5] + 97) % 26 + 97;
    }
    else
    {
      str2[v2] = (v1 - 39 - key[v3++ % v5] + 97) % 26 + 97;
    }
    if ( !(v3 % v5) )
      putchar(32);
    ++v2;
  }
}//通过分析可以知道，通过一系列变换后等于text就可以了
if ( !strcmp(text, str2) )
  puts("Congratulation!\n");
else
  puts("Try again!\n");
```

```
  return __readfsqword(0x28u) ^ v12;
}
```

c语言脚本

```c
#include<stdio.h>
int main()
{
 char key[] = "adsfkndcls";
 char text[] = "killshadow";
 int i;
 int v3=10;//长度
 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])
   {
    printf("%c",j);
    v3++;
    break;
   }
  }
 }
}
```

得到flag
KLDQCUDFZO