

【jarvisoj刷题之旅】逆向题目DDCTF - Hello的writeup

原创

iqiqiya 于 2018-09-09 20:06:13 发布 607 收藏

分类专栏: [我的逆向之路](#) [我的CTF之路](#) [我的CTF进阶之路](#) 文章标签: [【jarvisoj刷题之旅】逆向题目DDCTF - Hello DDCTF - Hello 逆向题目DDCTF - Hello的writeup reverse](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/xiangshangbashaonian/article/details/82561678>

版权



[我的逆向之路](#) 同时被 3 个专栏收录

108 篇文章 10 订阅

订阅专栏



[我的CTF之路](#)

92 篇文章 5 订阅

订阅专栏

[我的CTF进阶之路](#)

108 篇文章 18 订阅

订阅专栏

IDA64位载入

看到就几个函数 (没想到.jpg)

字符串都不用找

直接就可以确定sub_100000CE0就是关键函数

```
Library function Regular function Instruction Data Unexplored External symbol
Functions window IDA View-A Pseudocode-A Pseudocode-B Strings window Hex View-1
Function name Segn 1 int sub_100000CE0()
sub_100000C90 2 {
start 3 int result; // eax
sub_100000CE0 4 signed int v1; // [rsp+1Ch] [rbp-14h]
sub_100000DE0 5 int v2; // [rsp+24h] [rbp-Ch]
6
7 v2 = ((start - sub_100000C90) >> 2) ^ byte_100001040[0]; // start地址0000000100000CB0
// sub_100000C90地址0000000100000C90
8
9 result = sub_100000DE0();
10 if ( !(result & 1) )
11 {
12 v1 = 0;
13 while ( v1 < 55 )
14 {
15 byte_100001040[v1] -= 2;
16 byte_100001040[v1] ^= v2;
17 ++v1;
18 ++v2;
19 }
20 result = printf("\nFinal output is %s\n", &byte_100001040[1]);
21 }
22 return result;
23 }
```

<https://blog.csdn.net/xiangshangbashaonian>

分析就可以知道

先对两个函数的地址进行相减 再右移 与byte_100001040[0]进行异或

```
__text:0000000100000C90
__text:0000000100000C90 ; ===== S U B R O U T I N E =====
__text:0000000100000C90 ; Attributes: bp-based frame
__text:0000000100000C90 sub_100000C90 proc near ; CODE XREF: start+20↓p
__text:0000000100000C90 ; DATA XREF: sub_100000CE0+F↓o ...
__text:0000000100000C90 var_4 = dword ptr -4
__text:0000000100000C90
__text:0000000100000C91 push rbp
__text:0000000100000C94 mov rbp, rsp
__text:0000000100000C98 sub rsp, 10h
__text:0000000100000C9F lea rdi, aThisIsADummyFu ; "This is a dummy function\n"
__text:0000000100000CA1 mov al, 0
__text:0000000100000CA6 call _printf
__text:0000000100000CA9 mov [rbp+var_4], eax
__text:0000000100000CAD add rsp, 10h
__text:0000000100000CAE pop rbp
__text:0000000100000CAE sub_100000C90 retn
__text:0000000100000CAE endp
__text:0000000100000CAF ; -----
__text:0000000100000CAF align 10h https://blog.csdn.net/xiangshangbashaonian
__text:0000000100000CB0
```

```
__text:0000000100000CAF align 10h
__text:0000000100000CB0
__text:0000000100000CB0 ; ===== S U B R O U T I N E =====
__text:0000000100000CB0 ; Attributes: bp-based frame
__text:0000000100000CB0 public start
__text:0000000100000CB0 start proc near ; DATA XREF: sub_100000CE0+8↓o
__text:0000000100000CB0 ; sub_100000CE0+34↓o
__text:0000000100000CB0 var_8 = dword ptr -8
__text:0000000100000CB0 var_4 = dword ptr -4
__text:0000000100000CB0
__text:0000000100000CB1 push rbp
__text:0000000100000CB4 mov rbp, rsp
__text:0000000100000CB8 sub rsp, 10h
__text:0000000100000CBF lea rdi, aWelcome ; "Welcome\n"
__text:0000000100000CC6 mov [rbp+var_4], 0
__text:0000000100000CC8 mov al, 0
__text:0000000100000CC8 call _printf
__text:0000000100000CCD mov [rbp+var_8], eax
__text:0000000100000CD0 call sub_100000C90
__text:0000000100000CD5 xor eax, eax
__text:0000000100000CD7 add rsp, 10h
__text:0000000100000CDB pop rbp
__text:0000000100000CDC retn
__text:0000000100000CDC start endp https://blog.csdn.net/xiangshangbashaonian
```

后面就是一个循环 对byte_100001040[v1]进行运算

Py大法好:

```
PycharmProjects > reverse > jarvisoj > DD - Hello.py > DD - Hello.py x
1 a = [0x41, 0x10, 0x11, 0x11, 0x1B, 0x0A, 0x64, 0x67, 0x6A, 0x68,
2   0x62, 0x68, 0x6E, 0x67, 0x68, 0x6B, 0x62, 0x3D, 0x65, 0x6A,
3   0x6A, 0x3D, 0x68, 0x04, 0x05, 0x08, 0x03, 0x02, 0x02, 0x55,
4   0x08, 0x5D, 0x61, 0x55, 0x0A, 0x5F, 0x0D, 0x5D, 0x61, 0x32,
5   0x17, 0x1D, 0x19, 0x1F, 0x18, 0x20, 0x04, 0x02, 0x12, 0x16,
6   0x1E, 0x54, 0x20, 0x13, 0x14, 0x00, 0x00]
7 flag = ''
8 v2 = (0x0000000100000CB0 - 0x0000000100000C90) >> 2 ^ a[0]
9 for i in range(0,55):
10     a[i] -= 2
11     a[i] ^= v2
12     v2 = v2 + 1
13     flag += chr(a[i])
14 #print(flag)

Run: DD - Hello x
C:\Users\... \PycharmProjects\test\Scripts\python.exe "C:/Users/.../PycharmProjects/reverse/jarvisoj/DD - Hello.py"
DDCTF-5943293119a845e9bbdbde5a369c1f50@didichuxing.com

Process finished with exit code 0
```

<https://blog.csdn.net/xiangshangbashaonian>

DD - Hello

116 SOLVERS

100 REVERSE

Flag 是下一关的邮箱地址 (以 DD 开头) 。

1.Hello.12b9bde7c0c8558a9da42aa1798cafc8

DDCTF-5943293119a845e9bbdbde5a369c1f50@didichuxing.com

Correct Answer!!Congratulations!

<https://blog.csdn.net/xiangshangbashaonian>