XCTF reverse 高手进阶区 Mysterious



YQK易乾坤 ● 于 2020-11-18 14:56:41 发布 ● 122 ☆ 收藏
 分类专栏: XCTF REVERSE 高手进阶区 文章标签: 安全
 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/qq_30076719/article/details/109745451
 版权



XCTF REVERSE 高手进阶区 专栏收录该内容

1 篇文章 0 订阅 订阅专栏

这是一道基本的逆向题。。。

打开软件,然我们输入口令,猜测是输对了就能得到flag吧:

CTF	×
Input the pa	ssword: Crack

那就话不多说,直接放进ida啦

直接查看字符串,查看方法如下:



然后就能在字符串的窗口看到well done

🔀 🗄 IDA Vie	w-A 🔀 🗄	Pseud	ocode-C 🖾 🖹 Pseudocode-B 🕰 "" Strings window 🖾 🖹 Pseudoco
Address	Length	Туре	String
"" .text:0040D	00004275	С	<i>შიგები გაგებები გაგებები გაგებები გაგებები.</i>
"" .rdata:0042	A000000	С	well done
"" .rdata:0042	00000010	С	Buff3r_0v3rf 0w
"" .rdata:0042	0000000E	С	i386\\chkesp.c
"" .rdata:0042	000000DC	С	The value of ESP was not properly saved across a function call. This is usu
	00000011	<u> </u>	A CONTRACT OF A

双击打开



然后双击上图那里,打开对应的函数

f Functions window	E 🔛 🗄	🗐 IDA View-A 🔝 🗐 Pseudocode-C 💽	🕄 🗐 Pseudocode-B 🔝 '	Strings	window 🔝 🖃 Pseudoce	ode-A 🔀 🛗 Hex View-A 🔀 En	Enums 🔝 🗓
Function name	Segment			lea	edx, [ebp+Dest]		
👔 TimerFunc	.text			push	edx	; Dest	
DialogFunc	.text 🗏			call	_streat		
WinMain(x,x,x,x)	.text			add	esp, 8		
MinMain@16_0	.text			push	offset asc_422028	; "}"	
MialogFunc_0	.text			lea	eax, [ebp+Dest]		
TimerFunc_0	.text			push	eax	pest	
r chkesp	.text			call	_streat		
F strcpy	.text			auu 800	esp, 8		
Fstrcat	.text			nuch	a ·	иТиро	
🗿 atol	.text			nuch	offect Cantion	"well done"	
🗿 atoi	.text			lea	ery [ehn+Dest]	, well done	1
atoi64	.text			oush	ecx :	: loText	
🗿 _strien	.text			push	0	hWnd	
F memset	.text			call	ds:MessageBoxA		
🛐 start	.text			стр	esi, esp		
f sub_40199F	.text			call	chkesp		
🗿 sub 4019BA	.text						-
🛐amsg_exit	.text					-	
Ffast_error_exit	.text			MIL			
CrtDbgBreak	.text			I 나봄			
CrtSetReportMode	.text		10	a 1.049A	. .		
CrtSetReportFile	.text		10	U_4012H	n. ci ocn		
<pre>F]CrtDbgReport</pre>	.text		00	v e ch n	ffcot TimorFunc •	InTimerFunc	
🛐 _CrtMessageWindow	.text		pu pu	sh 3	F8h · i	Flanse	
J_isctype	.text 👻		pu pu	sh 1		IDEvent https://blog.csdn.n	
			Pu				

然后按下F5,查看伪代码(敲重点!!! 64位的按F5是没用的,必须要用32位的打开才有反应,我也被坑了,所以如果你看到 这里,你也恰巧打开的是64位的,那么请重新来一遍)

```
Ł
  if ( a3 == 1000 )
  {
     GetDlgItemTextA(hWnd, 1002, &Dst, 260);
     chkesp();
     strlen(&Dst);
     if ( strlen(&Dst) > 6 )
       ExitProcess(0);
                                                       T
     Value = atoi(&Dst) + 1;
     if ( Value == 123 && v14 == 120 && v16 == 122 && v15 == 121 )
     {
        strcpy((char *)Dest, 'flag');
        memset(&v9, 0, 0xFCu);
        v10 = 0;
        v11 = 0;
       _itoa(Value, &Source, 10);
strcat((char *)Dest, '{'');
strcat((char *)Dest, KSource)
strcat((char *)Dest, ''');

       strcat((char *)Dest, 'Buff3r_0v3rf|0w'
strcat((char *)Dest, '}');
MessageBoxA(0, Dest, 'well done'', 0);
abbuenc();
        _chkesp();
     }
     SetTimer(hWnd, 1u, 0x3E8u, (TIMERPROC)TimerFunc);
     _chkesp();
   з
  if ( a3 == 1001 )
  {
```

好了,看了代码之后,已经很明显了,框柱的连起来不就是我们平时看的flag格式嘛,那么只要得出绿色框中的的那个参数就可以了,于是打开_itoa函数,发现值为123,那么flag既可以拼接出来了,为:

flag{123_Buff3r_0v3rf|0w}



打开如下:

创作打卡挑战赛 赢取流量/现金/CSDN周边激励大奖