看雪CTF.TSRC 2018 团队赛 第一题 初世纪 writeup





CTF 专栏收录该内容

2 篇文章 0 订阅 订阅专栏 这个 writeup我花了1个多小时才写出来,我感觉这已经不是 writeup了,这简直就是解题图文教程!!

每个人都是从新手过来的,几年前我初学逆向的时候,看到大牛写的 writeup,一头雾水,根本看不懂,所以我 写这么详细是想让更多初学逆向的朋友能跟学会这门技术,感受她的魅力…

首先用PEID看一下有没有壳,发现没有壳,而且是个64位程序,这时候就可以排除用OD分析的可能了(很难过,最擅长的就是OD了),



因为CTF的题目一般不会有病毒后门什么的,所以直接在物理机上或者64位虚拟机里运行一下,猜猜思路



我输入序列号,发现会提示错误,但是我在IDA中搜索这个字符串,是什么都找不到的,这说明这个字符串可能 是个图片,emmm··但是我现在有2个思路了,可能会用到GetDlgItemTextA这个函数来获取字符串,还可能有 MessageBoxA这个函数来弹窗。所以直接导入IDA来看看

🔮 IDA - C:\Users\杨荣杰\Desktop\初世纪.exe

<u>F</u> ile	Ed	lit	Jum	р	Sea	rc <u>h</u>	⊻i	ew	Del	bugg	ger	<u>0</u>	ption	IS	Wi	ndc
1	-		⊨ -	•	-	₿ <u>₩</u>	₿ [™] T	-	4	1	1	-		0	1	1001
a 👎			11						11							
											100					-
-	Libr	ary	func	tior	-	Date	1 -	Reg	ılar	func	tio	n	Une	cplo	red	
f F	unct	i on	s win	ıdow										8	×	
Fun	ction	na	me												^	
f	sub_	140	0010	000												
f	sub_	140	0010	17C												
f	sub_	140	0010	DAD												
J f	sub_	140	0011	24												
f	Dialo	gF	unc	-												
f	sub_	140	0012	Έv		i										
f	sub_	140	0013	340												
f	sub_	140	0015	E0												
ţ f	sub_ cub	140	0016	04C			-									
f	sub_	140	0016	E0					-							
f	sub	140	0017	04						-	-					
f	sub_	140	0017	'B8												
f	sub_	140	0017	'C8												
f	sub_	140	0017	E4												
J f	sub_ sub	140	0017	174												
f	sub	140	0019	88												
f	sub_	140	0019	98												
f	start															
f	sub_	140	0019	D8												
ţ Ŧ	sub_	140	0014	AUC.												
f	sub_	140	0014													
f	sub	140	001E	90												
f	sub_	140	0010	CAO												
f	sub_	140	0010	010												
f	sub_	140	0016	:50												
J F	sub_	140	001E	AC												
f	sub	140	0016	C4												
f	sub_	140	001E	EC												
f	sub_	140	001F	04												
f	sub_	140	001F	68												
J f	sub_ sub	140	1001F	98 60												
f	sub_	140	001F	F8												
f	sub_	140	0020)C4												
f	sub_	140	0021	60												
f	sub_	140	0021	84												
ţ f	sub_	140	0021	80												
f	sub_	140	0022	250												
f	sub	140	0022	FC											¥	
<								leie	-	elene		leser		2	icie	0
Line	6 of	14	99													

看到这个函数,果断跟进去看看,这里交叉引用走到调用处

DA - C:\	Users\杨荣杰\	Desktop	初世纪.exe
----------	------------	---------	---------

〒UNA - CUUSERSWERENDERSUDERLOVERSUDERLOVER Fle Edit Jump Search View Debugger Options Windows Help 香品 キャチャー 物 物 物 物 物 体 1 gg 単 1 gg 通 ● 1 gg 通 ● 1 gg 通 ● 1 gg 通 ● 1 gg ■ 1 gg = 1 gg ■ 1 g

📝 Functions window	□ # ×	IDA View-A 🔀 🖸 Hex View-1 🔣 🖪 Structures 🖾 🗮	Enums 🖸 🎦 Imports 🚺
Function name	^	.text:000000014000129C call sub_140003A34	
F sub 140001000		• .text:00000001400012A1 du r5p, 400	
f sub 14000107C		• tavt - 0808080148081205 pop	
f sub_1400010A0		text:000000140001206 sub 14000120C endp	
f sub_140001124		text:0000000140001206	
f sub_14000120C		.text:00000001400012A6 ;	
1 DialogFunc		.text:0000000140001207 algn_140001207: ; DATA XREF: .pdata	:000000014005803010
f sub_1400012E0		• .text:00000001400012A7 align 10h	
f sub_140001340		.text:00000001400012B0	
5 sub_1400015E0		.text:00000001400012B0 ; S U B R O U T I N E	
f sub 14000164C		.text:0000000140001280	
7 sub 1400016E0			
f sub 140001704		.Text:0000000140001280; INI_PIK _Stocall VialogFunc(HWMU, UINI, WHKHM, LPHKMM)	3881948+E410
F sub_140001788		text - 0606060140601200	5883010
F sub_1400017C8		• .text: 0000000140001280 sub rsp. 28h	1000010
F sub_1400017E4		• .text:8888888148881284 cm edx. 111h	
f sub_1400017F8		- e .text: 🖾 xrefs to DialogFunc	- D X
f sub_140001974		• .text:	
5 sub_140001988		text: Direction Typ Address Text	
f start		🕂 🗠 🗠 🖂 text : 📴 Do o sub_140001340+F6 🛛 🗠 lea r9, DialogFunc : IpDialogFunc	
7 sub 1400019D8		- text - Do o .pdata:000000014005803C RUNTIME_FUNCTION <rva dialogfunc,="" rva="" sub_1400012<="" td=""><td>i0, \</td></rva>	i0, \
F sub 140001A0C		text	
7 sub_140001AE0		text	
🗲 sub_140001AF4		Lext: UK Lancel Search Help	
f sub_140001B90		• .text: Line 1 of 2	
f sub_140001CA0		.text:00000001400012D9 ;	···
f sub_140001D10		.text:00000001400012D9	
7 sub_140001ES0		.text:00000001400012D9 loc_1400012D9: ; CODE XREF: Dialog	unc+Aîj
f cub 140001E8C		.text:00000061400012D9 ; DialogFunc+157j	
7 sub 140001EC4		.text:00000000140001209 xor eax, eax	
7 sub 140001EEC		• .text:000000140001208 add r5p, 28n	
F sub 140001F04			
F sub_140001F68		tavt a a a a a a a a a a a a a a a a a a	
f sub_140001F98		Lext : 00000001400012E0	
f sub_140001FAC		.text:0000001400012E0 : SUBROUTINE	
f sub_140001FF8		.text:0000001400012E0	
f sub_1400020C4		.text:00000001400012E0	
✓ sub_1400020C4 ✓ sub_140002160		.text:00000001400012E0 .text:00000001400012E0 ; INT_PTRstdcall sub_1400012E0(HWND, UINT, WPARAM, LPARA	0
<pre>f sub_1400020C4 f sub_140002160 f sub_140002184 f sub_140002184 f sub_140002184</pre>		.text:0000000001400012E0 .text:00000001400012E0; INT_PTRstdcall sub_1400012E0(HWND, UINT, WPARAM, LPARA .text:00000001400012E0 sub_1400012E0 proc near ; DATA XREF: sub_14	() 1881348+10810 og ogda pel/ag . 44947909

Help	2										
* 's	* - * 🛋 🗙 👘	🝸 🕈 🚩 🗄	I II 🚡 🗩 🕀 :	h 🖨 谢							
ucti	on 📃 External sym	bol									
	IDA View-A		Hey View-1		A Structu	res 🖸	=	Enums			Imports
Ξř.	tout . 000000	04 6 00 04 6 5 5	JIER VIEW I		in Structu		لدفا	LICUNS	لما		Imports
1	toyt - 000000	014000145E 01500015E	loc 1000105F.			· CODE	VREE - cub	140001340+001	i		
4	text . 000000	014000145E	100_140001452.	mou	r9d 51h	: cchMa	water - Sub_	140001340.001	J		
	.text:000000	0140001464		lea	r8. [rbo+30h+	String] :	loString				
	.text:000000	0140001468		mov	edx, 3E8h	; nIDD1	qItem				
	.text:000000	014000146D		mov	rcx, rdi	; hDlg					
	.text:000000	0140001470		call	cs:GetDlgIter	TextA					
	.text:000000	0140001476		mov	r9d, 65h	; cchMa:	ĸ				
	.text:000000	014000147C		169	rø, [rsp+เงยเ	+nal in al i	ipstring				
	.text:000000	0140001481		mov	edx, 3E8h	; nIDD1	gItem	-	-		
	.text:000000	0140001486		MOV	rcx, rdi	; hDlg					
	.text:000000	0140001489		MOU	ebx, eax						
	.text:000000	014000148B		call	cs:GetDigiter	lexta					
	.text:000000	0140001491		cmp	eDX, 0	0041-54					
	.Lext:000000	0140001494 0160001494		JIIZ	SHUFL 100_146	DOIAFI					
	toyt-000000	0140001490 0150001508		Cub	eax, [rsprice	intoar_pol					
	text-000000	0140001470 014000140F		cmp	eav ehv						
	.text:000000	014000147E		inz	short loc 140	0014F1					
	.text:000000	01400014A2		movzx	eax. [rsp+130	h+var CF1					
	.text:000000	01400014A7		sub	eax, 40h						
	.text:000000	01400014AA		стр	eax, 5						
	.text:000000	01400014AD		jnz	short loc_140	0014F1					
	.text:000000	01400014AF		MOVZX	eax, [rsp+130	h+var_CE]					
	.text:000000	01400014B4		sub	eax, 70h						
	.text:000000	01400014B7		cmp	eax, 7						
	.text:000000	01400014BA		jnz	short loc_140	0014F1					
	.text:000000	01400014BC		MOVZX	eax, [rsp+130	h+var_CD]					
	.text:000000	0140001401		SUD	eax, own						
	.text:000000	0140001464		cmp	eax, y	0041-54					
	.Lext:000000	0140001467		JIIZ	SHUFL 100_140	0014F1					
	text-000000	01400014CF		SUD	eax 30h	n.oai_cc]					
	.text:000000	0140001402		CMD	eax, 9						
	.text:000000	01400014D4		inz	short loc 140	0014F1					
	.text:000000	01400014D6		MOVZX	eax, [rsp+130	h+var CB]					
	.text:000000	01400014DB		lea	rcx, String1						
	.text:000000	01400014E2		sub	eax, 40h						
	.text:000000	01400014E5		стр	eax, 8						
	.text:000000	01400014E8		jnz	short loc_140	0014F8					
	.text:000000	01400014EA		lea	rdx, [rsp+130	h+var_E8]					
	.text:000000	01400014EF		jmp	short loc_140	0014FD					
	.text:000000	01400014F1	;							-	
	.text:000000	0140001411	100 410004154-						÷.		
	.Lext:000000	0140001411	100_1400014F1:			; CUDE a	LAAA12LA.44	140001340+154 sati	U.		
-	text - 000000	01400014F7 01100011F7		103	Prev Steinet	, SUD_1	40001340+10 ina1				
		01400014111		TCG	ica, stringi	, that.	rug i				

看来我猜中了,的确是这个函数,这里可以大致的看到算法了,GetDlgltemTextA获取到文本之后,首先判断了 一下我们输入的字符串的位数是不是6位,如果不是就直接报错了,不会往下判断。我们按F5转换为伪代码看

```
• 73
         v8 = GetDlgItemTextA(v5, 1000, &String, 81);
         GetDlgItemTextA(v5, 1000, &v23, 101);
0 74
• 75
         if ( v8 != 6 || v23 != 54 || v24 != 69 || v25 != 119 || v26 != 105 || v27 != 57 )
  76
         {
• 77
           v9 = (CHAR *)&String1;
  78
         }
  79
         else
  80
         {
81
           v9 = (CHAR *)&String1;
           if ( u28 == 72 )
• 82
  83
           {
• 84
             v10 = (CHAR *)&v19;
  85 LABEL_19:
0 86
             lstrcpyA(v9, v10);
             DialogBoxParamA(hInstance, (LPCSTR)0x79, v5, sub_1400012E0, 0i64);
return sub_1400016E0((unsigned __int64)&v14 ^ v30);
0 87
• 88
  89
           }
  90
         }
91
         v10 = String2;
92
         goto LABEL_19;
  93
       }
94
       return sub_1400016E0((unsigned __int64)&v14 ^ v30);
• 95 }
```

因为OD不能用,所以还剩X64Debug和MDebug,MDebug太高端····我不太会用,但是 X64Debug和OD还是长 得挺像的,新手入门也快,这里就选 X64Debug了

首先载入创世纪.exe,我们有很多种方法来到算法处,第一个可以考虑用消息断点来到达关键处,但是我们前面用IDA分析出了GetDlgtemTextA这个函数,所以就直接ctrl+G到达定位到下图位置就可以了。

兼初世纪。	exe - PID: 1AA8	- 模块	:初世纪.exe - 缋	程:主线程	1988 - ×64dbg								
文件(<u>F</u>)	视图(Y) 调试((<u>D</u>) je	追踪(I) 插件(P)	收藏夹	(I) 选项(O) 帮助(H)	Oct 17	2018						
🖻 🕤 🔳		2	🔶 🦆 🌲	8 0	🗏 🎻 🥒 fx # 🛛 A2		9						
CPII	論 :古程图		= 「日 空行」	• 新石	- 内存布局 - 同语	田悦北		RHIT T) 22 년 10 년	⇔酒伴码	D 2168	₩ ₩	Snownon E
	T //611114		0000755712321	470	EE15 643E0400	Call	dword ntr ds	· [csGetD]		1	בתוכ ≃	- 7,51主	Sitowinality
			00007FF712321	476	41:B9 65000000	mov r	9d,65	. [sauces i	greenrexer	65:'e'			
		•	00007FF712321	47C	4C:8D4424 60	lea r	8,qword ptr	ss:[rsp+6	0]				
			00007FF712321	486	48:8BCF	mov e	cx.rdi						
			00007FF712321	489	8BD8	mov e	bx,eax			a			
		•	00007FF712321	48B	FF15 4F3E0400	call	qword ptr ds	::[<mark><&GetD]</mark>	gItemTextA>]			
			00007FF712321	491	83FB U6 75 50	ine A	的X,6 N冊約 ファテフィンス	21451					
			00007FF712321	496	0FB64424 60	movzx	eax.byte pi	r ss rsp	+601	把2绪eax			
	1		00007FF712321	49B	83E8 30	sub e	ax,30			2-30			
	1	•	00007FF712321	49E	3BC3	cmp e	ax,ebx			假码第一位	2次30和6比較	그는 가는 가 전에 위해	
	1		00007FF712321	4AU *	75 4F 0EP64434 61	Ine f	り世紀・/FF/123		+611	18:19:55 -11	10043UAN 6 FG 5X	,个为U则就	
			00007FF712321	4A7	83E8 40	sub e	ax.40		TOT I	假码第二位	2版40和5比较		
		٠	00007FF712321	4AA	83F8 05	cmp e	ax 5						
	[[•	00007FF712321	4AD ¥	75 42	jne 🕅	の世紀.7FF7123	214F1					
			00007FF712321	484	0FB64424 62 83F8 70	sub e	ax, byte p	in set fush	1762 J				
			00007FF712321	4B7	83F8 07	cmp e	ax,7						
	p	0	00007FF712321	4BA 🗸 🗸	75 35	jne 🕅	の世纪.7FF7123	214F1					
			00007FF712321	4BC	OFB64424 63	movzx	eax,byte pt	r ss:[rsp	+63				
			00007FF712321	401	83E8 60 83E8 09	cmp e	ax,60			9. '\+'			
	p	0	00007FF712321	407 .	75 28	ine A	0世纪.7FF7123	214F1		2. 10			
		٠	00007FF712321	409	OFB64424 64	movzx	eax,byte pt	r ss:[rsp	+64]				
		•	00007FF712321	4CE	83E8 30	sub e	ax,30			0.11+1			
	(₁		00007FF712321	404	75 1B	ine A	の世紀。7FF7123	214F1		2. (0			
			00007FF712321	4D6	OFB64424 65	movzx	eax, byte pt	r ss:[rsp	+65]	100000000000000000000000000000000000000			
		۰	00007FF712321	4DB	48:8D0D BE450500	lea r	cx, qword ptr	ds:[<mark>7FF7</mark>	12375AA0]	把可疑字符	5串地址给ecx		
			00007FF712321	4E2	83E8 40 97E9 09	sub e	ax,40						
	1 ₁		00007FF712321	4E8 ¥	75 OE	ine A	n世紀.7FF7123	214F8					
			00007FF712321	4EA	48:8D5424 48	lea r	dx, gword ptr	ss:[rsp+	48]	把可疑字??	5串地址给edx		
		•	00007FF712321	4EF ¥	EB OC	jmp 🕅	川世紀.7FF7123	214FD	122754407				
	L	>	00007FF712321	4F8	48:805424 30	lear	dx. gword ptr	SS Crsn+	30				
	L		00007FF712321	4FD	FF15 B53D0400	call	qword ptr ds	:[<mark><&lstrc</mark>	pyA>]	把ecx复制	到ed×		
			00007FF712321	503	48:8BOD B6450500	mov r	cx, qword ptr	ds [7FF7	12375AC0]				
		•	00007FF712321	50A	4C:8DOD CFFDFFFF	lea r	9,qword ptr	ds:[7FF71	23212E0]				
			00007FF712321	514	48:C74424 20 000000	mov r	word ptr ss:	[rsp+20].	0				
		•	00007FF712321	51D	BA 79000000	mov e	dx,79		-	79: 'y'			
			00007FF712321	522	FF15 D03D0400	call	qword ptr ds	:[<mark><&Dialo</mark>	gBoxParamA>]			
KIP		~	00007FF712321	528	58 01000000 F9 8600000	imp 2	ax,1 0冊約 7月月7122	21588					
			00007FF712321	532	45:33C9	xor r	9d,r9d	21500					
			00007FF712321	535	45:33C0	xor r	8d,r8d						
		•	00007FF712321	538	48:8BCF	mov r	cx,rdi		-1	-			
			00007FF712321	.53B	41:8051 10	l lea e	ux,qwora ptr	05:[r9+1	01	initias	e//initerenesse	impat/no	
1			<							11upo	managheee	numero a chelicit	

下面正式开始动态分析

我们在 GetDlgtemTextA函数处下断,运行程序,随便输入字符串230597,点击cheak,断在这里

文件(E) 视图(Y) 调武(D) 追踪(T) 插件(E) 收藏夹()	 (1) 法项(0) 報助(3) Oct 17 2018 (2) 参 か # 1 * 第 1 ■ ● 	
EPU ₩ 流程图 2 日志 1 笔记 ● 断点	■ 内存布局 调用堆栈 ■ SEP链 即本 ■ 符号 C FE15_4F3F0400 call guard ntr ds:[caGetDlgttemTextaal]	> 涤代码 22 引用 >> 线程 + 10 Snowman反编译器 ▲ 句柄 27 跟踪
00007FF71231451 00007FF71231451 00007FF71231496 00007FF71231496 00007FF71231496 00007FF71231496 00007FF71231496 00007FF71231496 00007FF71231496 00007FF71231496 00007FF71231496 00007FF71231446 00007FF71231447 00007FF71231447 00007FF71231446 00007FF71231446 00007FF71231467 00007FF71231467 00007FF71231467 00007FF71231467 00007FF71231468 00007FF71231468 00007FF71231468 00007FF71231468 00007FF71231468 00007FF71231468 00007FF71231468 00007FF71231468 00007FF71231468 000007FF71231468 00007FF71231468 00007FF71231468 00007FF71231511 00007FF71231511 00007FF71231518 00007FF71231518 00007FF71231518 00007FF71231518 00007FF71231518	By Brite Distriction Comp Bit (b) (b) (c) (c) (c) (c) (c) (c) (c) (c) (c) (c	With Comparison Compariso
		000000A497CFF000 500000000000000000000000000000000
emerging emerging	ASCIT ASCIT CC CC CC CC CC CC IIIIIIIIIIIIIIIIIIII	D0000004492/CFF008 F00000130010000 C00000492/CFF008 0000012000200 C00000492/CFF018 00000121EC24420 C00000492/CFF108 00018A010001AC8 C00000492/CFF108 0018A010001AC8 C00000492/CFF108 000001A13CE048803 C000000492/CFF108 0000014552983 C000000492/CFF108 00000100000000 C00000492/CFF108 00000100000000 C00000492/CFF108 00000000000000 C00000492/CFF108 00000000000000 C00000492/CFF108 00000000000000 C00000492/CFF108 00000000000000 C00000492/CFF108 00000000000000 C00000492/CFF108 000000000000000 C00000492/CFF108 000000000000000 C00000492/CFF108 0000000000000000000000000000000000
alt of a		

★ 初世纪.exe - PID: 1AC8 - 模块: 初世纪.exe - 线程: 主线程 2AB4 - x64dbg

然后继续向下F8单步走,先判断我们输入的是不是6位,然后把第一位数字2的ASCII提出来减去30(这里的30 是十六进制,下面也一样)和6进行比较,不一样就调到失败,这里我们就可以推出flag的第一位是 30+6=36(十六进制),由于比较结果不为0,jne是会跳转的,我们把ZF标志位改为1,让它不跳

00007FF712321491 S3FB 06 Cmp ebx, 6 判断我 00007FF712321494 × 75 5B jne 初世紀.7FF7123214F1 不是幼 00007FF712321496 0FB64424 60 movzx eax,byte ptr ss:[rsp+60] 控約 00007FF712321498 83E8 30 sub eax,30 2的AS 00007FF712321494 × 75 4F jne 初世紀.7FF7123214F1 優俏樂	新我们输入的长发是不是6 是夠昌接決到失敗了 :約ASCII十六进制治eax]ASCII十六进制-30 頁第一位减30和6比较 舀第一位减30和6比较,不为0则決转
---	--

继续向下分析,可以看出依次把第二、三、四、五、六位进相似的判断,我们依次逆向相加,推出第2-6位的 ASCII十六进制依次为45 77 69 39 48



判断完成后,程序对提示成功的字符串缓冲区地址进行复制等操作,最后当做参数传给这DialogBoxParamA这个函数

兼 初世纪.exe - PID: 1AC8 - 模块: 初世纪.exe - 线程: 主线程 2AB4 - x64dbg

文件(E) 视图(Y) 调试(D) 追踪(T) 插件(E) 收藏夹(I) 选项(D) 帮助(H) Oct 17 2018

	<i>⊘</i> ≒ <i>∞ ¶</i> JX <i>H</i> A₂		
📟 CPU 🏾 🥏 流程图 🛛 📄 知志 👘 笔记 🔹 🔹	断点 🛛 🛲 内存布局 🛛 🗐 调	用堆栈 🛛 🧠 SEH链 🛛 🖸 脚本 🛛 🛀 符号 🛛 🔇	> 源代码 🛛 🔎 引用 🛛 🛸 线程 🛛 🔞
• 00007FF71232148B	FF15 4F3E0400	<pre>call qword ptr ds:[<&GetDlgItemTextA>]</pre>	
00007FF712321491	83FB 06	cmp_ebx,6	判断我们输入的长度是不是6
• 00007FF712321494	✓ 75 5B	jne 初世纪 7FF7123214F1	小是新昌俊談到失敗了
00007FF712321496	OFB64424 60	movzx eax,byte ptr ss:[rsp+60]	把2的ASCII十六进刷路eax
• 00007FF/1232149B	8358 30	sub eax,30	2的ASCIITT八进刷-30 絶税1991-169時10和に比較
UUUU/FF/1232149E	3803	tomp eax,ebx	1181月第二1120月30月16日改
00007FF7123214A0	× 75 4F	JNE 初世紀、/FF/123214F1 Maryan any byte ptn cc. Encor(1)	181月第二120,30446143,个内0则被我
00007557122214A2	0750 40	Fub eax, byte ptr ss:[rspter]	(1864) 99 二次 95 4 0 第15 比較)
	0750 40	CMD COX,40	
00007FF712321400	× 75 42	ine 初世纪 7FF7123214F1	
00007FF7123214AF	0FB64424 62	movzy eax byte ntr ss:[rsn+62]	
• 00007FF7123214B4	8358 70	sub eax.70	
00007FF7123214B7	83F8 07	cmp eax.7	
00007FF7123214BA	× 75 35	ine 初世纪,7FF7123214F1	
00007FF7123214BC	0FB64424 63	movzx eax.byte ptr ss:[rsp+63]	
00007FF7123214C1	83E8 60	sub eax,60	CO0100000000000000000000000000000000000
00007FF7123214C4	83F8 09	cmp eax,9	9:'\t'
00007FF7123214C7	✓ 75 28	jne 初世纪.7FF7123214F1	
00007FF7123214C9	OFB64424 64	movzx eax,byte ptr ss:[rsp+64]	
00007FF7123214CE	83E8 30	sub eax,30	
00007FF7123214D1	83F8 09	cmp eax,9	9:'\t'
00007FF7123214D4	✓ 75 1B	jne 初世纪.7FF7123214F1	
• 00007FF7123214D6	OFB64424 65	movzx eax,byte ptr ss:[rsp+65]	
00007FF7123214DB	48:8DOD BE450500	[lea rcx, gword ptr ds:[7FF712375AA0]	把可融子符串地址给ecx
• 00007FF7123214E2	83E8 40	sub eax,40	
00007FF7123214E5	83F8 08	cmp eax,8	
000007FF/123214E8	✓ 75 UE 484.005.404.48	Jne 初世纪./FF/123214F8	如可將它來會接起來。也
00007FF7123214EA	48:805424 48	imn X0#f82 7EE7122214ED	忙的放于打甲地址站EUX
00007FF7123214E1		les new gwood ntn driffee71 276AAA	
L 00007FF7123214F1	40.0000 40450500	lea rdy gword ntr ss:[rsn+2]	
	FF15 B53D0400	call gword ntr ds:[calstron as]	把ecy復動剤edy
00007FF712321503	48:880D 86450500	mov nex gword ntr ds: [ZEEZ1 375AC0]	In conservation of the second s
• 00007FF71232150A	4C:8DOD CEEDEEEE	lea r9.gword ntr ds:[7EE71_8212E0]	
00007FF712321511	4C:8BC7	mov r8.rdi	
00007FF712321514	48:C74424 20 0000000	mov aword ptr ss: [rsp+20].	
00007FF71232151D	BA 79000000	mov edx 79	79: 'y'
RIP 00007FF712321522	FF15 D03D0400	<pre>call qword ptr ds:[<&DialogBoxParamA>]</pre>	
00007FF712321528	B8 01000000	mov eax,1	
• 00007FF71232152D	✓ E9 86000000	jmp 初世纪.7FF7123215B8	
00007FF712321532	45:3309	xor r9d,r9d	/plog.csdn.net/gg_41917908
00007FF712321535	45:33C0	xor r8d,r8d	

最后执行到这个call, CM弹出正确窗口

KCTF	_		\times
帮助(H) [TrackMe		
- 序列号: -			
230597	恭喜你!成功!		
看雪0		Exi	t

这时候我们就可以验证刚才的逆推结果了,我们把36 45 77 69 39 48转换为ASCII就是flag了



🗯 KCTF	<u></u>	
帮助(H)		_
- 序列号: -	TrackMe	_
6Ewi9H	恭喜你! 成功!	
看雪C		Exit
	UK	_

小白的 writeup,若有错误欢迎指正!!!