

VisualSite Designer破解教程

0x1, 次数破解:

1, 首先

在原文件中



有Number字符, 看来是因为次数会限制, 那么我们就先来破解它。

2, 由于无壳, 就直接上OD了:

004BD497	. E8 74000000	call VisualSi.004BD510	
004BD49C	. 8945 98	mov [local.26],eax	

下断点:

3, ctrl+F2, F9到断点, F7进入函数

004BD51C	. FF7424 10	push dword ptr ss:[esp+0x10]	Visua
004BD520	. E8 43000000	call <jmp.&MFC42.#1576>	
004BD525	. C2 1000	retn 0x10	

4, 老规矩, 继续:

0FC86241	8BCF	mov ecx,edi	
0FC86243	FFD6	call esi	
0FC86245	85C0	test eax,eax	

5, 再次就有点意思了, eax与1比较, 然后跳转, 这是关键所在, 严谨其间还是在进去一次吧;

00489912	. E8 132B0300	call <jmp.&MFC42.#2514>	
00489917	. 83F8 01	cmp eax,0x1	

6, 又找到了一个:

0FCB3507	50	push eax
0FCB3508	E8 53CFFEFF	call mfc42.#5718
0FCB350D	395F 20	cmp dword ptr ds:[edi+0x20],ebx

The screenshot shows a debugger window with a list of assembly instructions. The instruction at address 00489912 is highlighted in purple. A dialog box is open over the instruction list, titled '汇编于此处: 00489912' (Assemble here: 00489912). The dialog box contains a text input field with the instruction 'mov eax, 1'. Below the input field, there is a checkbox labeled '使用 NOP 填充' (Use NOP fill) which is checked. To the right of the checkbox are two buttons: '汇编' (Assemble) and '取消' (Cancel).

Address	Disassembly
00489903	lea ecx, dword ptr ss:[esp+0x20C]
0048990A	mov byte ptr ss:[esp+0x8778], 0xB
00489912	call <jmp.&MFC42.#2514>
00489917	cmp eax, 0x1
0048991A	je short MFC42Si.0048995C
0048991C	
0048991E	
00489923	
0048992A	
00489932	
00489937	
0048993E	mov byte ptr ss:[esp+0x8778], 0xC

汇编于此处: 00489912

mov eax, 1

☒ 使用 NOP 填充

汇编 取消

8. 广告弹窗:

9, 在OD中运行程序, 再关掉, 在OD中点击暂停:

然后进入堆栈段查看调用:

地址	堆栈	函数过程 / 参数	调用来自	结构
0019E9FC	76B48CE5	win32u.NtUserGetMessage	user32.76B48CDF	0019EA38
0019EA3C	0F5A3843	user32.GetMessageA	mfc42.0F5A383D	0019EA38
0019EA40	0058B264	pMsg = VisualSi.0058B264		
0019EA44	00000000	hWnd = NULL		
0019EA48	00000000	MsgFilterMin = 0x0		
0019EA4C	00000000	MsgFilterMax = 0x0		
0019EA60	0F5C056D	包含 mfc42.0F5A3843	mfc42.0F5C056B	0019EA5C
0019EA8C	0F5D350D	mfc42.#5718	mfc42.0F5D3508	0019EA88
0019EAD6	00480C29	? <jmp.&MFC42.#2514>	VisualSi.00480C24	0019EAD4
0019EB48	0F5A83D5	包含 VisualSi.00480C29	mfc42.0F5A83D3	0019EB4C
0019EB50	0F59F73B	mfc42.0F5A83C0	mfc42.0F59F736	0019EB4C
0019EBF0	0F59FE40	包含 mfc42.0F59F73B	mfc42.0F59FE3E	0019EBEC
0019EC1C	0F59A44B	包含 mfc42.0F59FE40	mfc42.0F59A449	0019EC18
0019EC9C	0F59A2FF	mfc42.#1109	mfc42.0F59A2FA	0019EC98
0019ECE4	76B5BF1B	包含 mfc42.0F59A2FF	user32.76B5BF19	0019ECE0
0019ED10	76B583EA	user32.76B5BEF0	user32.76B583E5	0019ED0C
0019EDF8	76B57F8A	user32.76B58040	user32.76B57F85	0019EDF4
0019EE5C	76B5A6D9	包含 user32.76B57F8A	user32.76B5A6D7	0019EE58
0019EE9C	7795CD3D	包含 user32.76B5A6D9	ntdll.7795CD3B	0019EE98

在所选位置，我们看到其他都是mfc的系统调用，只有所选位置是程序自身的，点击进去相应位置：

00480C1C	. C74424 68 00	mov dword ptr ss:[esp+0x68],0x0
00480C24	. E8 01B80300	call <jmp.&MFC42.#2514>
00480C29	. 8D4C24 00	lea ecx,dword ptr ss:[esp]
00480C2D	. C74424 68	
00480C35	. E8 DEBA03	
00480C3A	. 8B4C24 60	
00480C3E	. 64:890D 0	
00480C45	. 83C4 6C	
00480C48	. C3	
00480C49	. 90	

汇编于此处: 00480C24

☒ 使用 NOP 填充

nop填充，保存文件。

这后，就没有广告了。