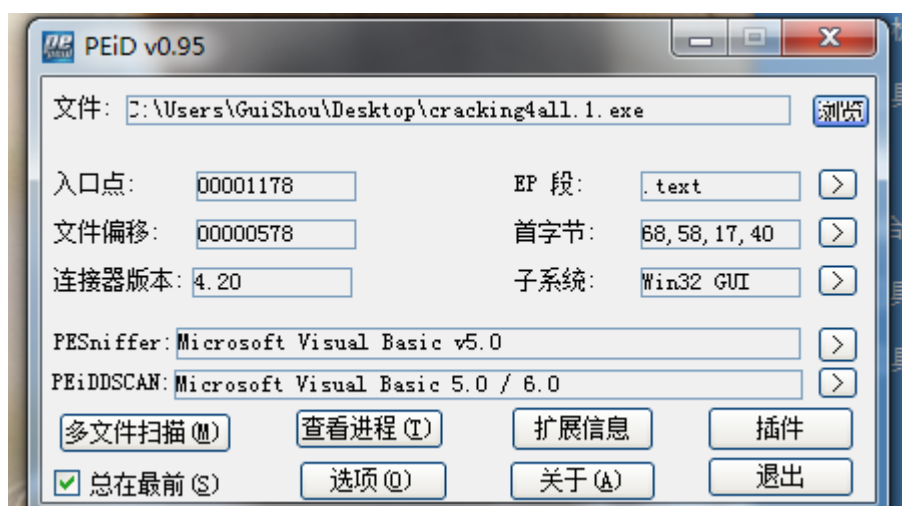


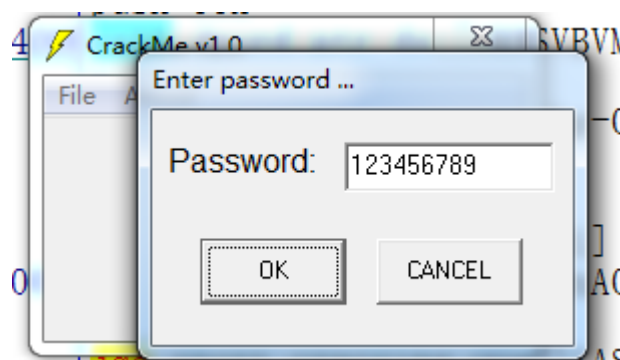
查壳
分析程序
算法分析
写出注册机
校验结果

查壳



目标程序是VB写的，序列号保护方式，难度为一颗星

分析程序



随便输入一个序列号，来到字符串的错误提示处

文件(F) 查看(V) 调试(D) 插件(P) 选项(T) 窗口(W) 帮助(H) [+]						快速搜索 Tools BreakPoint						[+] 1 e m t w h c p k b r ... []					
---	--	--	--	--	--	-----------------------	--	--	--	--	--	---	--	--	--	--	--

找到跳转到错误提示的地方 跟过去

地址	HEX	数据	反汇编	注释	寄存器 (FPU)	
00403322	0F80 3E020000		ja cracking.00403566		AX 0018F400	
00403328	8BF8		mov edi,eax		CX 0018F370	
0040332A	E9 C0FEFFFF		jmp cracking.004031EF		DX 0057FA98	
0040332F	8D45 C8		lea eax,dword ptr ss:[ebp-0x38]		EX 0F05A825 msvbvm50.__vbaStr	
00403332	8D8D 38FFFFFF		lea ecx,dword ptr ss:[ebp-0xC8]		FP 0018F30C	
00403338	50		push eax	var18 = 0018F400	BP 0018F438	
00403339	51		push ecx	var28 = 0018F370	SI 00000003	
0040333A	C785 40FFFFFF		mov dword ptr ss:[ebp-0xC0],cracking.004027C8	qBQSYdXUe_B\	DI 0000000F	
00403344	C785 38FFFFFF		mov dword ptr ss:[ebp-0xC8],0x8008			
00403346	FF15 44614000		call dword ptr ds:[&MSVBVM50.__vbaVarTstEq]	__vbaVarTstEq	P 0040334E cracking.0040334E	
00403354	66:85C0		test ax,ax		0 ES 002B 32位 0(FFFFFFFF)	
00403357	B9 04000280		mov ecx,0x80020004		0 CS 0023 32位 0(FFFFFFFF)	
0040335C	B8 0A000000		mov eax,0xA		0 SS 002B 32位 0(FFFFFFFF)	
00403361	894D 80		mov dword ptr ss:[ebp-0x80],ecx		0 DS 002B 32位 0(FFFFFFFF)	
00403364	8985 78FFFFFF		mov dword ptr ss:[ebp-0x88],eax		0 FS 0053 32位 7EFD0000(FFF	
0040336A	894D 90		mov dword ptr ss:[ebp-0x70],ecx		0 GS 002B 32位 0(FFFFFFFF)	
0040336D	8945 88		mov dword ptr ss:[ebp-0x78],eax		0	
00403370	0F84 E8000000		ja cracking.0040345E		0 LastErr ERROR_SUCCESS (00	
00403376	8B35 9C614000		mov esi,dword ptr ds:[&MSVBVM50.__vbaVarDup]	msvbvm50.__vbaVarDup	FL 00000202 (NO,NB,NE,A,NS,PO	
0040337C	BF 08000000		mov edi,0x8		0 empty 0.0	
00403381	8D95 28FFFFFF		lea edx,dword ptr ss:[ebp-0xD8]		1 empty 0.0	
00403387	8D4D 98		lea ecx,dword ptr ss:[ebp-0x68]		2 empty 0.0	
0040338A	C785 30FFFFFF		mov dword ptr ss:[ebp-0xD0],cracking.00402824	Valid	3 empty 0.0	
00403394	89BD 28FFFFFF		mov dword ptr ss:[ebp-0xD8],edi		4 empty 0.0	
004027C8=cracking.004027C8 (UNICODE "qBQSYdXUe_BV")						
堆栈 ss:[0018F370]=004027C8 (cracking.004027C8), UNICODE "qBQSYdXUe_BV"						
地址	HEX	数据	UNICODE	地址	数值	注释
0057FAC4	03 00 01 00	01 00 01 00 03 00 01 00 01 00 10 00	qBQSYdXUe_B\	0018F30C	0018F370	var28 = 0018F370
0057FAD4	00 00 6F 00	72 00 6C 00 64 00 00 00 CA 7F 11 4C	World. 蝴蝶	0018F310	0018F400	var18 = 0018F400
0057FAE4	00 FF 00 88	02 00 00 00 30 00 00 00 30 00 30 00	蝴蝶.0.0	0018F314	0018F444	
0057FAF4	00 00 00 00	01 00 01 00 03 00 01 00 01 00 01 00	..	0018F318	0018F520	
0057FB04	00 00 00 00	F7 7F 11 4C 00 00 00 88 02 00 00 00	.. 蝴蝶.蝴蝶	0018F31C	002A6ABC	
0057FB14	03 00 00 00	30 00 30 00 00 00 01 00 01 00 01 00	L.00.	0018F320	00000031	
0057FB24	02 00 01 00	00 00 5C 00 00 00 00 00 50 7F 11 4C	L.1.蝴蝶	0018F324	00000001	
0057FB34	00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00		0018F328	00000015	
0057FB44	00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00		0018F32C	FFFFFFFF	
0057FB54	00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00		0018F330	0000000E	
0057FB64	00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00		0018F334	75AD1898	返回到 user32.75AD1898
0057FB74	00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00		0018F338	B7012391	
0057FB84	00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00		0018F33C	00000000	
0057FB94	00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00		0018F340	00000000	
0057FBA4	00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00		0018F344	009AFCD0	UNICODE "OK"

接着发现了一个VB的比较函数，下断点，观察一下堆栈，参数一和参数二如下：

地址	HEX	数据	UNICODE	地址	数值	注释
004027C8	71 00 42 00	51 00 53 00 59 00 64 00 58 00 55 00	qBQSYdXUe_B\	0018F30C	0018F370	var28 = 0018F370
004027D8	65 00 5F 00	42 00 5C 00 56 00 00 00 36 00 00 00	e_B\	0018F310	0018F400	var18 = 0018F400
004027E8	50 00 61 00	73 00 73 00 77 00 6F 00 72 00 64 00	Password	0018F314	0018F444	
004027F8	20 00 63 00	6F 00 72 00 72 00 65 00 63 00 74 00	correct	0018F318	0018F520	
00402808	2C 00 20 00	68 00 65 00 68 00 65 00 2C 00 20 00	, hehe,	0018F31C	002A6ABC	
00402818	3A 00 2D 00	29 00 00 00 0A 00 00 00 56 00 61 00	(-)...Va	0018F320	00000031	
00402828	6C 00 69 00	64 00 00 00 23 3D FB FC FA A0 68 10	lid. 漢口	0018F324	00000001	
00402838	A7 38 08 00	2B 33 71 B5 22 3D FB FC FA A0 68 10	与C. 许	0018F328	00002015	
00402848	A7 38 08 00	2B 33 71 B5 02 00 00 00 30 28 40 00	与C. 许	0018F32C	FFFFFFFF	
00402858	40 28 40 00	00 00 00 00 50 00 00 00 50 00 61 00	.. P. Pa	0018F330	0000000E	
00402868	73 00 73 00	77 00 6F 00 72 00 64 00 20 00 69 00	ssword i	0018F334	75AD1898	返回到 user32.75AD1898
00402878	6E 00 63 00	6F 00 72 00 72 00 65 00 63 00 74 00	ncorrect	0018F338	B7012391	
00402888	2C 00 20 00	70 00 6C 00 65 00 61 00 73 00 65 00	, please	0018F33C	00000000	
00402898	20 00 74 00	72 00 79 00 20 00 61 00 67 00 61 00	try aga	0018F340	00000000	
004028A8	69 00 6E 00	20 00 2E 00 2E 00 2E 00 00 00 00 00	in	0018F344	009AFCD0	UNICODE "OK"

HEX 数据														UNICODE		地址		数值		注释																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
00585264	03	00	01	00	01	00	01	00	03	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00	01	00

算法分析

首先获取序列号长度，以序列号长度为大循环的循环次数，然后还有一个小循环的循环次数为4

然后从序列号第一位开始取1个字符，取字符的位置每次+1，取字符的最大起始位置为序列号的长度，即大循环的循环次数

地址	HEX 数据	反汇编	注释	寄存器 (FPU)	
00403217	51	push ecx	Start = 0x18F3A0	EAX 0018F3C0	
00403218	8D45 98	lea eax,dword ptr ss:[ebp-0x68]		ECX 0018F3A0	
0040321B	52	push edx	dString8 = 00000001	EDX 00000001	
0040321C	50	push eax	RetBUFFER = 0018F3C0	EBX 0F05A825 msvbvm50. __vbaStrVarVal	
0040321D	C745 B0 0100	mov dword ptr ss:[ebp-0x50],0x1		ESP 0018F304	
00403224	C745 A8 0200	mov dword ptr ss:[ebp-0x58],0x2		EBP 0018F438	
0040322B	FF15 38614000	call dword ptr ds:[&MSVBVM50.#rtcmidCharVar_632]	从序列号开始的位置取1个字符	ESI 00000001	
00403231	B8 02000000	mov eax,0x2	eax=2	EDI 00000001	
00403236	8D8D 78FFFFFF	lea ecx,dword ptr ss:[ebp-0x88]		EIP 00403263 cracking. 00403263	
0040323C	0FBFD6	movsx edx,si	edx=1	C 1 ES 002B 32位 0(FFFFFFFF)	
0040323F	8985 78FFFFFF	mov dword ptr ss:[ebp-0x88],eax		P 0 CS 0023 32位 0(FFFFFFFF)	
00403245	8945 88	mov dword ptr ss:[ebp-0x78],eax		A 1 SS 002B 32位 0(FFFFFFFF)	
00403248	51	push ecx	Length8 = 0x18F3A0	Z 0 DS 002B 32位 0(FFFFFFFF)	
00403249	8D45 88	lea eax,dword ptr ss:[ebp-0x78]	eax=2	S 1 FS 0053 32位 7EFD0000(FFF)	
0040324C	52	push edx	Start = 0x1	T 0 GS 002B 32位 0(FFFFFFFF)	
0040324D	8D8D 68FFFFFF	lea ecx,dword ptr ss:[ebp-0x98]		D 0	
00403253	50	push eax	dString8 = 0018F3C0	O 0 LastErr ERROR_SUCCESS (00000000)	
00403254	51	push ecx	RetBUFFER = 0018F3A0	EFL 00000293 (NO,B,NE,BE,S,PO,L,LE)	
00403255	C745 80 0100	mov dword ptr ss:[ebp-0x80],0x1		ST0 empty 0.0	
0040325C	C745 90 D007	mov dword ptr ss:[ebp-0x70],0x7D0		ST1 empty 0.0	
00403263	FF15 38614000	call dword ptr ds:[&MSVBVM50.#rtcmidCharVar_632]	从硬编码的字符串中取一个字符串	ST2 empty 0.0	
00403269	8D55 98	lea edx,dword ptr ss:[ebp-0x68]		ST3 empty 0.0	
0040326C	8D45 C0	lea eax,dword ptr ss:[ebp-0x40]		ST4 empty 0.0	
0040326F	52	push edx	edx="1"->Serial[0]	ST5 empty 0.0	
ds:[00406138]-0F 04386E (msvbvm50.#rtcmidCharVar)					
地址	HEX 数据	UNICODE	地址	数值	注释
0018F3C0	02 00 00 00	83 24 04 26	0018F304	0018F3A0	RetBUFFER = 0018F3A0
0018F3D0	08 00 18 00	38 EC 0D 0F	0018F308	0018F3C0	dString8 = 0018F3C0
0018F3E0	02 00 00 00	30 00 00 00	0018F30C	00000001	Start = 0x1
0018F3F0	00 00 00 00	00 00 00 00	0018F310	0018F3B0	Length8 = 0x18F3B0
0018F400	00 00 00 00	00 00 00 00	0018F314	0018F444	
0018F410	08 00 00 00	30 00 00 00	0018F318	0018F520	
0018F420	B4 F4 18 00	84 F9 18 00	0018F31C	01E76ABC	
0018F430	18 10 40 00	01 00 00 00	0018F320	00000002	

然后从硬编码的字符串02 00 00 00第一位取一个字符，开始的位置每次+1，取字符的最大起始位置为4，即上面的小循环的循环次数，超过4则重新回到1

地址	HEX 数据	反汇编	注释	寄存器 (FPU)
0040323C	0FBFD6	movsx edx,si	edx=1	EAX 00000031
0040323F	8985 78FFFFFF	mov dword ptr ss:[ebp-0x88],eax		ECX 0018F307
00403245	8945 88	mov dword ptr ss:[ebp-0x78],eax		EDX 0054FAC6
00403248	51	push ecx	Length8 = 0x18F307	EBX 0F05A825 msvbvm50. __vbaStrVarVal
00403249	8D45 88	lea eax,dword ptr ss:[ebp-0x78]	eax=2	ESP 0018F314
0040324C	52	push edx	Start = 0x54FAC6	EBP 0018F438
0040324D	8D8D 68FFFFFF	lea ecx,dword ptr ss:[ebp-0x98]		ESI 00000001
00403253	50	push eax	dString8 = 00000031	EDI 00000001
00403254	51	push ecx	RetBUFFER = 0018F307	EIP 0040327A cracking. 0040327A
00403255	C745 80 0100	mov dword ptr ss:[ebp-0x80],0x1		C 1 ES 002B 32位 0(FFFFFFFF)
0040325C	C745 90 D007	mov dword ptr ss:[ebp-0x70],0x7D0		P 1 CS 0023 32位 0(FFFFFFFF)
00403263	FF15 38614000	call dword ptr ds:[&MSVBVM50.#rtcmidcharvar_632]	从硬编码的字符串中取一个字符串	A 1 SS 002B 32位 0(FFFFFFFF)
00403269	8D55 98	lea edx,dword ptr ss:[ebp-0x68]		Z 0 DS 002B 32位 0(FFFFFFFF)
0040326C	8D45 C0	lea eax,dword ptr ss:[ebp-0x40]		S 1 FS 0053 32位 7EFD0000(FFF)
0040326F	52	push edx	edx="1"->Serial[0]	T 0 GS 002B 32位 0(FFFFFFFF)
00403270	50	push eax		D 0
00403271	FFD3	call ebx	取出Key[0]的字符串指针	O 0 LastErr ERROR_SUCCESS
00403273	50	push eax	String = 00000031 ???	EFL 00000297 (NO,B,NE,BE,S,PO,L,LE)
00403274	FF15 0C614000	call dword ptr ds:[&MSVBVM50.#rtcmidcharvar_516]	将序列号第一位转成ASCII->31	ST0 empty 0.0
0040327A	0FBFD6	movsx edx,ax	edx=31	ST1 empty 0.0
0040327D	8D8D 68FFFFFF	lea ecx,dword ptr ss:[ebp-0x98]		ST2 empty 0.0
00403283	8D45 BC	lea eax,dword ptr ss:[ebp-0x44]		ST3 empty 0.0
00403286	51	push ecx		ST4 empty 0.0
00403287	50	push eax		ST5 empty 0.0
00403288	8995 E8FFFFFF	mov dword ptr ss:[ebp-0x118],edx		
0040328E	FFD3	call ebx	取出"2"的字符串指针	
00403290	50	push eax	String = 00000032 ???	
00403291	FF15 0C614000	call dword ptr ds:[&MSVBVM50.#rtcmidcharvar_516]	将"2"转成ASCII->32	
00403297	8B95 E8FFFFFF	mov edx,dword ptr ss:[ebp-0x118]	edx=Key[0]	
0040329D	0FBFC8	movsx ecx,ax	ecx="2"	
004032A0	33D1	xor edx,ecx	edx="2"*Key[0]	
004032A2	8D85 58FFFFFF	lea eax,dword ptr ss:[ebp-0xA8]		
004032A8	52	push edx		
004032A9	50	push eax		
004032AA	FF15 64614000	call dword ptr ds:[&MSVBVM50.#rtcmidcharvar_608]	msvbvm50.rtcVarBstrFromAnsi	
004032B0	8D4D C8	lea ecx,dword ptr ss:[ebp-0x38]		
004032B3	8D95 58FFFFFF	lea edx,dword ptr ss:[ebp-0xA8]		
004032B9	51	push ecx		
004032BA	8D85 48FFFFFF	lea eax,dword ptr ss:[ebp-0xB8]		
004032C0	52	push edx		
ax=0032				

然后将取出的那一位序列号转成ASCII值

地址	HEX 数据	反汇编	注释	寄存器 (FPU)
00403271	FFD3	call ebx	取出Key[0]的字符串指针	EAX 00000032
00403273	50	push eax	String = 00000032 ???	ECX 0018F307
00403274	FF15 0C614000	call dword ptr ds:[&MSVBVM50.#rtcmidcharvar_516]	将序列号第一位转成ASCII->31	EDX 0054FAC6
0040327A	0FBFD6	movsx edx,ax	edx=31	EBX 0F05A825 msvbvm50. __vbaStrVarVal
0040327D	8D8D 68FFFFFF	lea ecx,dword ptr ss:[ebp-0x98]		ESP 0018F314
00403283	8D45 BC	lea eax,dword ptr ss:[ebp-0x44]		EBP 0018F438
00403286	51	push ecx		ESI 00000001
00403287	50	push eax		EDI 00000001
00403288	8995 E8FFFFFF	mov dword ptr ss:[ebp-0x118],edx		EIP 00403297 cracking. 00403297
0040328E	FFD3	call ebx	取出"2"的字符串指针	C 1 ES 002B 32位 0(FFFFFFFF)
00403290	50	push eax	String = 00000032 ???	P 1 CS 0023 32位 0(FFFFFFFF)
00403291	FF15 0C614000	call dword ptr ds:[&MSVBVM50.#rtcmidcharvar_516]	将"2"转成ASCII->32	A 1 SS 002B 32位 0(FFFFFFFF)
00403297	8B95 E8FFFFFF	mov edx,dword ptr ss:[ebp-0x118]	edx=Key[0]	Z 0 DS 002B 32位 0(FFFFFFFF)
0040329D	0FBFC8	movsx ecx,ax	ecx="2"	S 1 FS 0053 32位 7EFD0000(FFF)
004032A0	33D1	xor edx,ecx	edx="2"*Key[0]	T 0 GS 002B 32位 0(FFFFFFFF)
004032A2	8D85 58FFFFFF	lea eax,dword ptr ss:[ebp-0xA8]		D 0
004032A8	52	push edx		O 0 LastErr ERROR_SUCCESS (00000000)
004032A9	50	push eax		EFL 00000297 (NO,B,NE,BE,S,PE,L,LE)
004032AA	FF15 64614000	call dword ptr ds:[&MSVBVM50.#rtcmidcharvar_608]	msvbvm50.rtcVarBstrFromAnsi	ST0 empty 0.0
004032B0	8D4D C8	lea ecx,dword ptr ss:[ebp-0x38]		ST1 empty 0.0
004032B3	8D95 58FFFFFF	lea edx,dword ptr ss:[ebp-0xA8]		ST2 empty 0.0
004032B9	51	push ecx		ST3 empty 0.0
004032BA	8D85 48FFFFFF	lea eax,dword ptr ss:[ebp-0xB8]		ST4 empty 0.0
004032C0	52	push edx		ST5 empty 0.0
ax=0032				

再将取出的硬编码字符串转成ASCII值

地址	HEX 数据	反汇编	注释	寄存器 (FPU)
0040326F	52	push edx	edx="1"→Serial[0]	EAX 00000032
00403270	50	push eax		ECX 00000032
00403271	FFD3	call ebx	取出Key[0]的字符串指针	EDX 00000031
00403273	50	push eax	String = 00000032 ???	EBX 0F05A825 msvbvm50.__vbaStrVarVa
00403274	FF15 0C614000	call dword ptr ds:[&MSVBVM50.#rtcAnsiValueBstr_516>]	将序列号第一位转成ASCII→31	ESP 0018F314
0040327A	0FBFD0	movsx edx, ax	edx=31	EBP 0018F438
0040327D	8D8D 68FFFFFF	lea ecx, dword ptr ss:[ebp-0x98]		ESI 00000001
00403283	8D45 BC	lea eax, dword ptr ss:[ebp-0x44]		EDI 00000001
00403286	51	push ecx		EIP 004032A0 cracking.004032A0
00403287	50	push eax		C 1 ES 002B 32位 0(FFFFFFFF)
00403288	8995 E8FEFFFF	mov dword ptr ss:[ebp-0x118], edx		P 1 CS 0023 32位 0(FFFFFFFF)
0040328E	FFD3	call ebx	取出"2"的字符串指针	A 1 SS 002B 32位 0(FFFFFFFF)
00403290	50	push eax	String = 00000032 ???	Z 0 DS 002B 32位 0(FFFFFFFF)
00403291	FF15 0C614000	call dword ptr ds:[&MSVBVM50.#rtcAnsiValueBstr_516>]	将"2"转成ASCII→32	S 1 FS 0053 32位 7EFD0000(FFF)
00403297	8B95 E8FEFFFF	mov edx, dword ptr ss:[ebp-0x118]	edx=Key[0]	T 0 GS 002B 32位 0(FFFFFFFF)
0040329D	0FBFC8	movsx ecx, ax	ecx="2"	D 0
004032A0	33D1	xor edx, ecx	edx="2"~Key[0]	0 0 LastErr ERROR_SUCCESS (00000000)
004032A2	8D85 58FFFFFF	lea eax, dword ptr ss:[ebp-0xA8]		EFL 00000297 (NO, B, NE, BE, S, PE, L, LE)
004032A8	52	push edx		ST0 empty 0.0
004032A9	50	push eax		ST1 empty 0.0
004032AA	FF15 64614000	call dword ptr ds:[&MSVBVM50.#rtcVarBstrFromAnsi_608>]	msvbvm50.rtcVarBstrFromAnsi	ST2 empty 0.0
004032B0	8D4D C8	lea ecx, dword ptr ss:[ebp-0x38]		ST3 empty 0.0
004032B3	8D95 58FFFFFF	lea edx, dword ptr ss:[ebp-0xA8]		ST4 empty 0.0
004032B9	51	push ecx		ST5 empty 0.0
004032BA	8D85 48FFFFFF	lea eax, dword ptr ss:[ebp-0xB8]		
004032C0	52	push edx		
004032C1	50	push eax		
004032C2	FF15 70614000	call dword ptr ds:[&MSVBVM50.__vbaVarCat>]	拼接异或后的字符串	
004032C8	8BD0	mov edx, eax		
004032CA	8D4D C8	lea ecx, dword ptr ss:[ebp-0x38]		
004032CD	FF15 F8604000	call dword ptr ds:[&MSVBVM50.__vbaVarMove>]	将计算后的结果移动到[ebp-0x38]	
004032D3	8D4D BC	lea ecx, dword ptr ss:[ebp-0x44]		
004032D6	8D55 C0	lea edx, dword ptr ss:[ebp-0x40]		
004032D9	51	push ecx		
004032DA	52	push edx		
004032DB	6A 02	push 0x2		
004032DD	FF15 8C614000	call dword ptr ds:[&MSVBVM50.__vbaFreeStrList>]	msvbvm50.__vbaFreeStrList	
004032E3	83C4 0C	add esp, 0xC		
004032E6	8D85 58FFFFFF	lea eax, dword ptr ss:[ebp-0xA8]		
004032EC	8D8D 68FFFFFF	lea ecx, dword ptr ss:[ebp-0x98]		

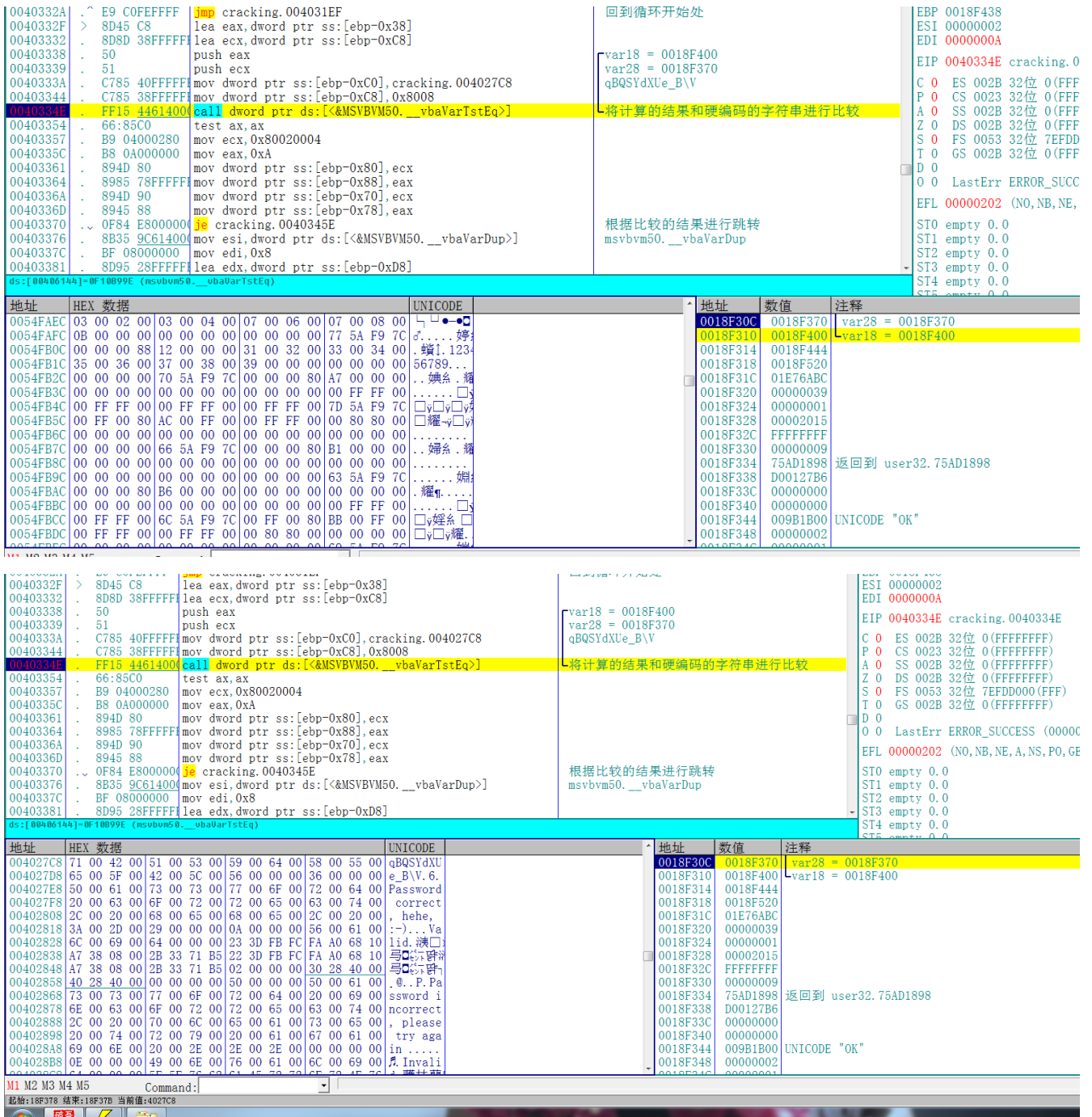
最后将取出的序列号的ASCII值和取出的硬编码的ASCII值进行异或

地址	HEX 数据	反汇编	注释	寄存器 (FPU)
004032A0	33D1	xor edx, ecx	edx="2"~Key[0]	EAX 0018
004032A2	8D85 58FFFFFF	lea eax, dword ptr ss:[ebp-0xA8]		ECX 0018
004032A8	52	push edx		EDX 0018
004032A9	50	push eax		EBX 0F05
004032AA	FF15 64614000	call dword ptr ds:[&MSVBVM50.#rtcVarBstrFromAnsi_608>]	msvbvm50.rtcVarBstrFromAnsi	ESP 0018
004032B0	8D4D C8	lea ecx, dword ptr ss:[ebp-0x38]		EBP 0018
004032B3	8D95 58FFFFFF	lea edx, dword ptr ss:[ebp-0xA8]		ESI 0000
004032B9	51	push ecx		EDI 0000
004032BA	8D85 48FFFFFF	lea eax, dword ptr ss:[ebp-0xB8]		EIP 0040
004032C0	52	push edx		C 0 ES
004032C1	50	push eax		P 1 CS
004032C2	FF15 70614000	call dword ptr ds:[&MSVBVM50.__vbaVarCat>]	拼接异或后的字符串	A 0 SS
004032C8	8BD0	mov edx, eax		Z 1 DS
004032CA	8D4D C8	lea ecx, dword ptr ss:[ebp-0x38]		S 0 FS
004032CD	FF15 F8604000	call dword ptr ds:[&MSVBVM50.__vbaVarMove>]	将计算后的结果移动到[ebp-0x38]	T 0 GS
004032D3	8D4D BC	lea ecx, dword ptr ss:[ebp-0x44]		D 0
004032D6	8D55 C0	lea edx, dword ptr ss:[ebp-0x40]		0 0 Las
004032D9	51	push ecx		EFL 0000
004032DA	52	push edx		ST0 empt
004032DB	6A 02	push 0x2		ST1 empt
004032DD	FF15 8C614000	call dword ptr ds:[&MSVBVM50.__vbaFreeStrList>]	msvbvm50.__vbaFreeStrList	ST2 empt
004032E3	83C4 0C	add esp, 0xC		ST3 empt
004032E6	8D85 58FFFFFF	lea eax, dword ptr ss:[ebp-0xA8]		ST4 empt
004032EC	8D8D 68FFFFFF	lea ecx, dword ptr ss:[ebp-0x98]		ST5 empt

拼接字符串，将异或后的结果保存到[ebp-0x38]这个位置，

地址	HEX 数据	反汇编	注释
004032FA	8D45 88	lea eax, dword ptr ss:[ebp-0x78]	
004032FD	52	push edx	
004032FE	8D4D 98	lea ecx, dword ptr ss:[ebp-0x68]	
00403301	50	push eax	
00403302	8D55 A8	lea edx, dword ptr ss:[ebp-0x58]	
00403305	51	push ecx	
00403306	52	push edx	
00403307	6A 06	push 0x6	
00403309	FF15 00614000	call dword ptr ds:[&MSVBVM50.__vbaFreeVarList>]	msvbvm50.__vbaFreeVarList
0040330F	83C4 1C	add esp, 0x1C	
00403312	66:46	inc si	i++
00403314	B8 01000000	mov eax, 0x1	
00403319	66:03C7	add ax, di	
0040331C	0F80 44020000	jo cracking.00403566	
00403322	0F80 3E020000	jo cracking.00403566	
00403328	8BF8	mov edi, eax	
0040332A	E9 C0FEFFFF	jmp cracking.004031EF	回到循环开始处
0040332F	8D45 C8	lea eax, dword ptr ss:[ebp-0x38]	

然后回到循环开始处开始新一轮循环，所以只要观察[ebp-0x38]，就能看到最后的结果



最后将用户名异或得出的计算结果和硬编码的字符串进行比较，根据比较的结果提示正确与否

写出注册机

整理一下这个程序的算法，这个程序的算法其实就是一元一次方程，即 $Serial \wedge 02\ 00\ 00$

$00(Key) = qBQSYdXUe_B \setminus V(result)$

Serial是未知的，我们可以根据已知的result去异或已知Key得到正确的序列号，这个可以手动计算，也可以写个循环计算，代码如下：

```
#include <iostream>
using namespace std;

int main()
```

```

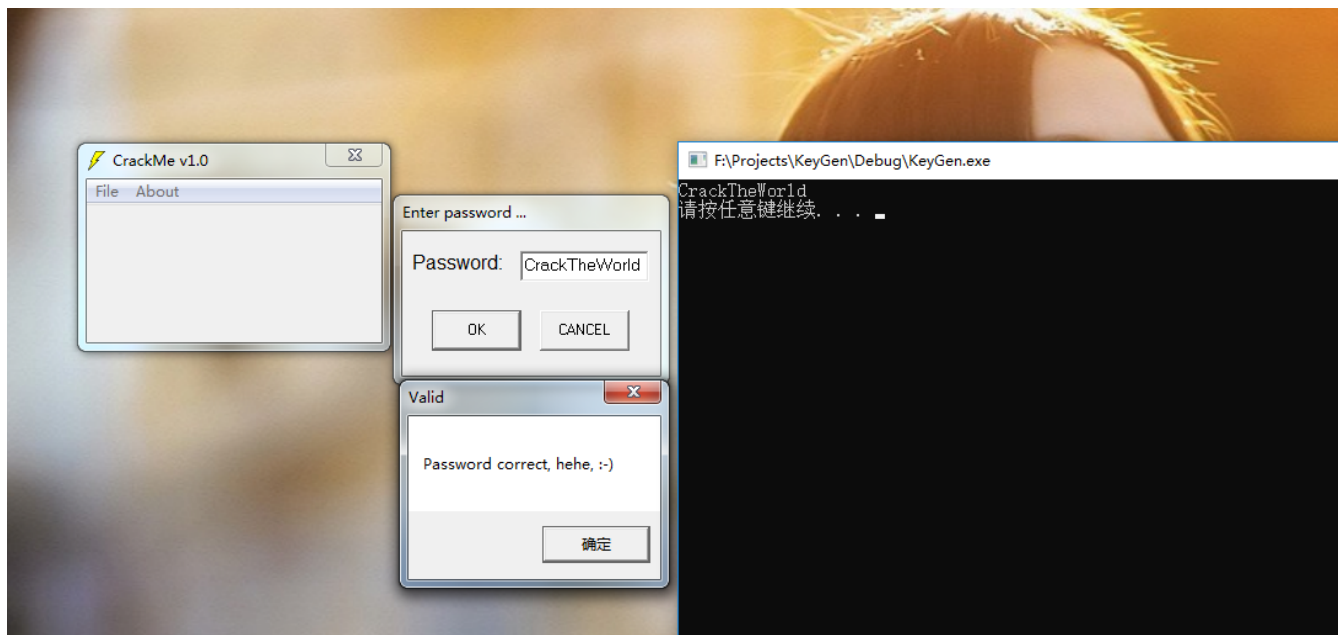
{
    char serial[13] = { 0 };
    char result[14] = { "qBQSYdXue_B\\V" };
    char key[17] = { "2000200020002000" };

    for (int i = 0; i < 14; i++)
    {
        serial[i] = result[i] ^ key[i];
    }
    serial[13] = 0;
    printf("%s\n", serial);
    system("pause");
    return 0;
}

```

校验结果

输入计算出来的序列号，提示正确，破解完成



需要相关文件可以到我的Github下载: <https://github.com/TonyChen56/160-Crackme>