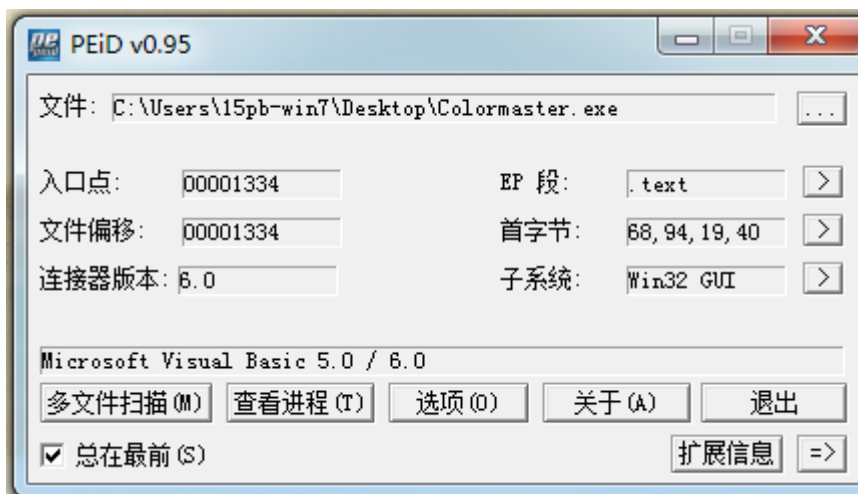


- 查壳
- 分析程序
- 算法分析
 - 基础校验
- 第一部分
- 第二部分
- 第三部分
- 第四部分
- 第五部分
- 第六部分
- 写出注册机
- 验证结果

查壳



这个Crackme也是用VB写的，难度是问号，自我感觉应该值三颗星

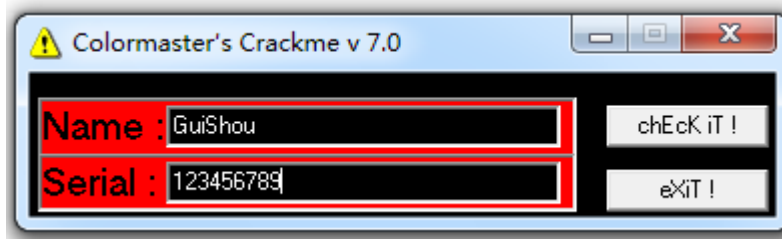
分析程序

地址	HEX	数据	反汇编	注释
0040371E	.	8985 5C	mov dword ptr ss:[ebp-0xA4],eax	kernel32.BaseThreadInitThunk
00403724	.	8985 6C	mov dword ptr ss:[ebp-0x94],eax	kernel32.BaseThreadInitThunk
0040372A	.	C785 14	mov dword ptr ss:[ebp-0xEC],Colormas.00401F2C	Colormaster's Crackme 7.0
00403734	.	89BD 0C	mov dword ptr ss:[ebp-0xF4],edi	
0040373A	.	FFD6	call esi	<&MSVBVM60.__vbaVarDup>
0040373C	.	8D95 1C	lea edx,dword ptr ss:[ebp-0xE4]	
00403742	.	8D4D 8C	lea ecx,dword ptr ss:[ebp-0x74]	
00403745	.	C785 24	mov dword ptr ss:[ebp-0xDC],Colormas.00401F80	Gratulation ,du hast es geschafft!
0040374F	.	89BD 1C	mov dword ptr ss:[ebp-0xE4],edi	
00403755	.	FFD6	call esi	
00403757	.	8D85 5C	lea eax,dword ptr ss:[ebp-0xA4]	
0040375D	.	8D8D 6C	lea ecx,dword ptr ss:[ebp-0x94]	
00403763	.	50	push eax	kernel32.BaseThreadInitThunk
00403764	.	8D95 7C	lea edx,dword ptr ss:[ebp-0x84]	
0040376A	.	51	push ecx	
0040376B	.	52	push edx	Colormas.<ModuleEntryPoint>
0040376C	.	8D45 8C	lea eax,dword ptr ss:[ebp-0x74]	
0040376F	.	6A 40	push 0x40	
00403771	.	50	nush eax	kernel32.BaseThreadInitThunk

首先根据字符串提示，来到按钮点击事件的开头

00402B0F	CC	int3	
00402B10	> 55	push ebp	按钮点击事件开头
00402B11	. 8BEC	mov ebp,esp	
00402B13	. 83EC 0C	sub esp,0xC	
00402B16	. 68 C611	push <jmp.&MSVBVM60.__vbaExceptHandler>	SE 处理程序安装
00402B1B	. 64:A1 0	mov eax,dword ptr fs:[0]	
00402B21	. 50	push eax	kernel32.BaseThreadInitThunk
00402B22	. 64:8925	mov dword ptr fs:[0],esp	
00402B29	. 81EC A4	sub esp,0x1A4	
00402B2F	. 53	push ebx	
00402B30	. 56	push esi	
00402B31	. 57	push edi	
00402B32	. 8965 F4	mov dword ptr ss:[ebp-0xC],esp	
00402B35	. C745 F8	mov dword ptr ss:[ebp-0x8],Colormas.00401108	
00402B3C	. 8B75 08	mov esi,dword ptr ss:[ebp+0x8]	
00402B3F	. 8BC6	mov eax,esi	
00402B41	. 83E0 01	and eax,0x1	
00402B44	. 8945 FC	mov dword ptr ss:[ebp-0x4],eax	kernel32.BaseThreadInitThunk

下断点，随便输入一个用户名和序列号，开始分析程序



算法分析

这个程序的算法分成六个部分，下面依次对每个部分的算法进行详细的讲解

基础校验

00402CAB	. 50	push eax	
00402CAC	. FF15 34104000	call dword ptr ds:[<&MSVBVM60.__vbaHresultCheckObj>]	msvbvm60.__vbaHresultCheckObj
00402CB2	> 8B55 D8	mov edx,dword ptr ss:[ebp-0x28]	
00402CB5	. 52	push edx	
00402CB6	. FF15 10104000	call dword ptr ds:[<&MSVBVM60.__vbaLenBstr>]	String = "GuiShou" 求用户名长度
00402CBC	. 33C9	xor ecx,ecx	
00402CBE	. 83F8 04	cmp eax,0x4	用户名长度必须大于4
00402CC1	. 0f9ec1	setle cl	
00402CC4	. F7D9	neg ecx	
00402CC6	. 66:898D DCFEFFFF	mov word ptr ss:[ebp-0x124],cx	
00402CCD	. 8D4D D8	lea ecx,dword ptr ss:[ebp-0x28]	
00402CD0	. FF15 F0104000	call dword ptr ds:[<&MSVBVM60.__vbaFreeStr>]	msvbvm60.__vbaFreeStr
00402CD6	. 8D4D B8	lea ecx,dword ptr ss:[ebp-0x48]	
00402CD9	. FF15 F4104000	call dword ptr ds:[<&MSVBVM60.__vbaFreeObj>]	msvbvm60.__vbaFreeObj
00402CDF	. 66:399D DCFEFFFF	cmp word ptr ss:[ebp-0x124],bx	
00402CE6	. 74 0F84 B0000000	je Colormas.00402D9C	
00402CEC	. 8B35 D4104000	mov esi,dword ptr ds:[<&MSVBVM60.__vbaVarDup>]	msvbvm60.__vbaVarDup
00402CF2	. B9 04000280	mov ecx,0x80020004	

首先获取输入的用户名的长度，必须大于4，如果不大于则提示错误

第一部分

最难也是最麻烦的就是第一部分了，如果解决了这个部分，那么后面剩余的四个部分就不成问题

00402DF1	8B00 4B	mov eax, dword ptr ss:[ebp-0x20]	push edx	String = ""	EAX 0012F324
00402DF4	52		push ecx	求用户名长度	ECX 0012F334
00402DF7	FF15 10104000	call dword ptr ds:[<MSVBVM60. __vbaLenBstr>]			EDX 0012F478
00402DFB	8985 14FFFFFF	mov dword ptr ss:[ebp-0xEC], eax			EBX 00000000
00402E01	8D85 1CFFFFFF	lea eax, dword ptr ss:[ebp-0xE4]			ESP 0012F2C0
00402E07	8D8D 0CFFFFFF	lea ecx, dword ptr ss:[ebp-0xF4]			EBP 0012F49C
00402E0D	50	push eax			ESI 0023C568 UNICODE "为"
00402E0E	8D95 FCFFFFFF	lea edx, dword ptr ss:[ebp-0x104]			EDI 72A19FF1 msbvm60. __vbaObjSet
00402E14	51	push ecx			EIP 00402E46 Colormas.00402E46
00402E15	8D85 88FFFFFF	lea eax, dword ptr ss:[ebp-0x178]			C 0 ES 0023 32位 0(FFFFFFFF)
00402E1B	52	push edx			P 0 CS 001B 32位 0(FFFFFFFF)
00402E1C	8D8D 98FFFFFF	lea ecx, dword ptr ss:[ebp-0x168]			A 0 SS 0023 32位 0(FFFFFFFF)
00402E22	50	push eax			Z 0 DS 0023 32位 0(FFFFFFFF)
00402E23	8D55 DC	lea edx, dword ptr ss:[ebp-0x24]			S 0 FS 003B 32位 7FFFFFFF(FFF)
00402E26	51	push ecx			T 0 GS 0000 NULL
00402E27	52	push edx			D 0
00402E28	C785 0CFFFFFF 03	mov dword ptr ss:[ebp-0xF4], 0x3			0 0 LastErr ERROR_SUCCESS (00000000)
00402E32	C785 04FFFFFF 01	mov dword ptr ss:[ebp-0xFC], 0x1			EFL 00000202 (NO, NB, NE, A, NS, PO, GE, G)
00402E3C	C785 FCFFFFFF 02	mov dword ptr ss:[ebp-0x104], 0x2			ST0 empty 0.0
00402E40	FF15 3C104000	call dword ptr ds:[<MSVBVM60. __vbaObjSet>]		以用户名长度为循环次数	ST1 empty 0.0
00402E4C	8D4D B8	lea ecx, dword ptr ss:[ebp-0x28]			ST2 empty 0.0
00402E4F	8985 7CFFFFFF	mov dword ptr ss:[ebp-0x184], eax			ST3 empty 0.0

地址	HEX 数据	反汇编	UNICODE	地址	数值	注释
00405000	00 00 00 00	00 00 00 00	00 00 00 00	0012F2C0	0012F478	Counter8 = 0012F478
00405010	68 C5 23 00	00 00 00 00	00 00 00 00	0012F2C4	0012F334	TMStep8 = 0012F334
00405020	00 00 00 00	00 00 00 00	00 00 00 00	0012F2C8	0012F324	TMStep8 = 0012F324
00405030	02 00 00 00	00 00 00 00	00 00 00 00	0012F2CC	0012F398	Start8 = 0012F398
00405040	41 00 43 00	3A 00 5C 00	50 00 72 00	0012F2D0	0012F3A8	End8 = 0012F3A8
00405050	72 00 61 00	6D 00 6D 00	65 00 5C 00	0012F2D4	0012F3B8	Step8 = 0012F3B8
00405060	63 00 72 00	6F 00 73 00	6F 00 66 00	0012F2D8	0012F4A8	
00405070	56 00 69 00	73 00 75 00	61 00 6C 00	0012F2DC	0012F578	UNICODE "I"

1-1 首先获取用户名长度，然后以用户名长度作为循环的次数，开始一轮循环，即i=strlen(username)

地址	HEX 数据	反汇编	注释	寄存器 (FPU)
00402E9F	68 A0000000	push 0xA0		EAX 00000047
00402EA4	68 941E4000	push Colormas.00401E94		ECX 0012F2D7
00402EA9	52	push edx		EDX 0023931E UNICODE
00402EAA	50	push eax		EBX 00000000
00402EAB	FF15 34104000	call dword ptr ds:[<MSVBVM60. __vbaHResultCheckObj>]	msbvm60. __vbaHResultCheckObj	ESP 0012F2D8
00402EB1	8B45 D4	mov eax, dword ptr ss:[ebp-0x2C]		EBP 0012F49C
00402EB4	50	push eax		ESI 0023C568 UNICODE
00402EB5	FF15 24104000	call dword ptr ds:[<MSVBVM60. #516>]	String = 00000047 ???	EDI 72A19FF1 msbvm60
00402EBB	8B0E	mov ecx, dword ptr ds:[esi]	Colormas.004052F8	EIP 00402EBB Colormas
00402EBD	56	push esi		C 1 ES 0023 32位 0(F)
00402EBE	8985 E8FFFFFF	mov dword ptr ss:[ebp-0x118], eax	保存username[0]的ASCII值	P 1 CS 001B 32位 0(F)
00402EC4	FF91 28030000	call dword ptr ds:[ecx+0x328]		A 1 SS 0023 32位 0(F)
00402ECA	8D55 B0	lea edx, dword ptr ss:[ebp-0x50]		Z 0 DS 0023 32位 0(F)
00402ECD	50	push eax		S 1 FS 003B 32位 7(F)
00402ECE	52	push edx		T 0 GS 0000 NULL
00402ECF	FFD7	call edi	msbvm60. __vbaObjSet	D 0
00402ED1	8985 D4FFFFFF	mov dword ptr ss:[ebp-0x12C], eax		0 0 LastErr ERROR_SU
00402ED7	8B06	mov eax, dword ptr ds:[esi]	Colormas.004052F8	EFL 00000297 (NO, B, NI
00402ED9	56	push esi		ST0 empty 0.0
00402EDA	FF90 24030000	call dword ptr ds:[eax+0x324]		ST1 empty 0.0
00402EE0	8D4D B8	lea ecx, dword ptr ss:[ebp-0x48]		ST2 empty 0.0
00402EE3	50	push eax		ST3 empty 0.0

1-2 获取用户名第一位的ASCII值->username[0]

地址	HEX 数据	反汇编	注释	寄存器 (FPU)
00402F05	68 A41E4000	push Colormas.00401EA4		EAX 00000000
00402F0A	52	push edx		ECX 00000047
00402F0B	50	push eax		EDX 0012F428
00402F0C	FF15 34104000	call dword ptr ds:[<MSVBVM60. __vbaHResultCheckObj>]	msbvm60. __vbaHResultCheckObj	EBX 00000000
00402F12	8B45 D8	mov eax, dword ptr ss:[ebp-0x28]		ESP 0012F2D4
00402F15	50	push eax		EBP 0012F49C
00402F16	FF15 B0104000	call dword ptr ds:[<MSVBVM60. __vbaR8Str>]	把432.4转为浮点数	ESI 0023C568 UNICODE
00402F1C	0FBF8D E8FFFFFF	movsx ecx, word ptr ss:[ebp-0x118]	ecx=username[0]的ASCII值	EDI 72A19FF1 msbvm60
00402F23	898D 74FFFFFF	mov dword ptr ss:[ebp-0x18C], ecx	保存username[0]到[ebp-0x18C]	EIP 00402F63 Colormas
00402F29	8D55 8C	lea edx, dword ptr ss:[ebp-0x74]		C 0 ES 0023 32位 0(F)
00402F2C	DB85 74FFFFFF	fld dword ptr ss:[ebp-0x18C]	将username[0]压入ST0	P 1 CS 001B 32位 0(F)
00402F32	52	push edx		A 0 SS 0023 32位 0(F)
00402F33	C785 0CFFFFFF 05	mov dword ptr ss:[ebp-0xF4], 0x5		Z 1 DS 0023 32位 0(F)
00402F3D	C745 94 15000000	mov dword ptr ss:[ebp-0x6C], 0x15		S 0 FS 003B 32位 7(F)
00402F44	C745 8C 02000000	mov dword ptr ss:[ebp-0x74], 0x2		T 0 GS 0000 NULL
00402F4B	DD9D 6CFFFFFF	fstp qword ptr ss:[ebp-0x194]		D 0
00402F51	DC8D 6CFFFFFF	fmul qword ptr ss:[ebp-0x194]	[ebp-0x194]=username[0]	0 0 LastErr ERROR_SU
00402F57	DC0D 00114000	fmul qword ptr ds:[0x401100]	ST0=username[0]*432.4	EFL 00000246 (NO, NB, E
00402F5D	DD9D 14FFFFFF	fstp qword ptr ss:[ebp-0x194]	保存结果到[ebp-0xEC]->546160.1156	ST0 empty 0.0
00402F63	DFF0	fstsw ax		ST1 empty 0.0
00402F65	A8 0D	test al, 0xD		ST2 empty 0.0
00402F67	0F85 0E090000	jnz Colormas.0040387B		ST3 empty 0.0

地址	64 位双精度	地址	数值	注
0012F3B0	546160.1159999999	0012F2D4	0012F428	
0012F3C0	2.635739459941128e-308	0012F2D8	0012F4A8	
0012F3D0	0.0	0012F2DC	0012F578	UN
0012F3E0	9.677007007000874e-307	0012F2E0	00000001	
0012F3F0	-NAN FFFFFFFF 00002015	0012F2E4	75A7B01C	us
0012F400	3.564631367539919e-315	0012F2E8	00000001	
0012F410	2.121995795905928e-314	0012F2EC	75A282EF	返
0012F420	-NAN FFFFFFFF 00000001	0012F2F0	00000001	

1-3 用户名的第一位的ASCII值乘以432.4再乘以17.79，-> username[0]*432.4*17.79

地址	HEX 数据	反汇编	注释	寄存器 (FPU)
00402F73	50	push eax		EAX 0023FA24 UNICODE "36410.6744"
00402F74	FF15 C0104000	call dword ptr ds:[<&MSVBVM60.#573>]	msvbvm60.rtcHexVarFromVar	ECX 0012F46C
00402F7A	88BD D4FEFFFF	mov ecx,dword ptr ss:[ebp-0x12C]		EDX 00000000
00402F80	8D95 0CFEFFFF	lea edx,dword ptr ss:[ebp-0xF4]		EBX 01AE2E48
00402F86	8D85 7CFEFFFF	lea eax,dword ptr ss:[ebp-0x84]		ESP 0012F2D8
00402F8C	52	push edx		EBP 0012F49C
00402F8D	8B19	mov ebx,dword ptr ds:[ecx]		ESI 0023C568 UNICODE "劲@"
00402F8F	8D8D 6CFEFFFF	lea ecx,dword ptr ss:[ebp-0x94]		EDI 72A19FF1 msbvm60.__vbaObjSet
00402F95	50	push eax		EIP 00402FA8 Colormas.00402FA8
00402F96	51	push ecx		C 0 ES 0023 32位 0 (FFFFFFFF)
00402F97	FF15 98104000	call dword ptr ds:[<&MSVBVM60.__vbaVarDiv>]	var28 = 0023FA24 SaveToSI = 0012F46C __vbaVarDiv	P 1 CS 001B 32位 0 (FFFFFFFF)
00402F9D	8D55 D0	lea edx,dword ptr ss:[ebp-0x30]	计算的结果/0x15	A 0 SS 0023 32位 0 (FFFFFFFF)
00402FA0	50	push eax		Z 1 DS 0023 32位 0 (FFFFFFFF)
00402FA1	52	push edx		S 0 FS 003B 32位 7FFDF000 (FFF)
00402FA2	FF15 A0104000	call dword ptr ds:[<&MSVBVM60.__vbaStrVarVal>]	ARG2 = NULL __vbaStrVarVal	T 0 GS 0000 NULL
00402FA8	89DD 68FEFFFF	mov dword ptr ss:[ebp-0x198],ebx	将浮点数的结果转为字符串	D 0
00402FAE	8B9D D4FEFFFF	mov ebx,dword ptr ss:[ebp-0x12C]		O 0 LastErr ERROR_SUCCESS (00000000)
00402FB4	50	push eax		EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
00402FB5	8B85 68FEFFFF	mov eax,dword ptr ss:[ebp-0x198]		ST0 empty 0.0
00402FBB	53	push ebx		ST1 empty 0.0
00402FBC	FF50 54	call dword ptr ds:[eax+0x54]		ST2 empty 0.0
00402FBF	85C0	test eax,ecx		ST3 empty 0.0

1-4 将1-3-result除以0x15, 然后把浮点数值转为字符串-> username[0]*432.4*17.79/0x15

地址	HEX 数据	反汇编	注释	寄存器 (FPU)
00403065	FF15 60104000	call dword ptr ds:[<&MSVBVM60.#599>]	msvbvm60.rtcSendKeys	EA
0040306B	8D4D 8C	lea ecx,dword ptr ss:[ebp-0x74]		EC
0040306E	FF15 0C104000	call dword ptr ds:[<&MSVBVM60.__vbaFreeVar>]	msvbvm60.__vbaFreeVar	ED
00403074	8D8D 88FEFFFF	lea ecx,dword ptr ss:[ebp-0x178]		EB
0040307A	8D95 98FEFFFF	lea edx,dword ptr ss:[ebp-0x168]		ES
00403080	51	push ecx		EB
00403081	8D45 DC	lea eax,dword ptr ss:[ebp-0x24]		ES
00403084	52	push edx		ED
00403085	50	push eax		
00403086	FF15 E8104000	call dword ptr ds:[<&MSVBVM60.__vbaVarForNext>]	__vbaVarForNext	EI
0040308C	8985 7CFEFFFF	mov dword ptr ss:[ebp-0x184],eax		
00403092	33DB	xor ebx,ebx		
00403094	E9 CBFDFEFFFF	jmp Colormas.00402E64	跳到循环开始处	
00403099	8B0E	mov ecx,dword ptr ds:[esi]		
0040309B	56	push esi		
0040309C	FF91 20030000	call dword ptr ds:[ecx+0x320]	Colormas.004052F8	

1-5 跳到循环开始处, 开始新一轮循环

地址	HEX 数据	反汇编	注释	寄存器 (FPU)
004030DD	68 A41E4000	push Colormas.00401EA4		EAX 0023FA74 UNICODE "56923"
004030E2	52	push edx		ECX 0012F470
004030E3	50	push eax		EDX 0023FA74 UNICODE "56923"
004030E4	FF15 34104000	call dword ptr ds:[<&MSVBVM60.__vbaHResultCheckObj>]	msvbvm60.__vbaHResultCheckObj	EBX 01AE2E48
004030EA	8B85 DCFEFFFF	mov eax,dword ptr ss:[ebp-0x124]		ESP 0012F2D8
004030F0	8B4D D8	mov ecx,dword ptr ss:[ebp-0x28]		EBP 0012F49C
004030F3	51	push ecx	ecx=最后一次的结果	ESI 0023C568 UNICODE "劲@"
004030F4	8B18	mov ebx,dword ptr ds:[eax]		EDI 72A19FF1 msbvm60.__vbaObjSet
004030F6	FF15 B0104000	call dword ptr ds:[<&MSVBVM60.__vbaR8Str>]		EIP 00403113 Colormas.00403113
004030FC	FF15 64104000	call dword ptr ds:[<&MSVBVM60.__vbaFPFix>]	msvbvm60.__vbaFPFix	C 0 ES 0023 32位 0 (FFFFFFFF)
00403102	83EC 08	sub esp,0x8	对计算的结果舍去小数部分	P 1 CS 001B 32位 0 (FFFFFFFF)
00403105	DD1C24	fstp qword ptr ss:[esp]		A 0 SS 0023 32位 0 (FFFFFFFF)
00403108	FF15 7C104000	call dword ptr ds:[<&MSVBVM60.__vbaStrR8>]		Z 1 DS 0023 32位 0 (FFFFFFFF)
0040310E	8BD0	mov edx,ecx		S 0 FS 003B 32位 7FFDF000 (FFF)
00403110	8D4D D4	lea ecx,dword ptr ss:[ebp-0x2C]		T 0 GS 0000 NULL
00403113	FF15 DC104000	call dword ptr ds:[<&MSVBVM60.__vbaStrMove>]	将舍去小数后的结果保存到[ebp-0x2C]	D 0
00403119	8BD3	mov edx,ebx		O 0 LastErr ERROR_SUCCESS (00000000)
0040311B	8B9D DCFEFFFF	mov ebx,dword ptr ss:[ebp-0x124]		EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
00403121	50	push eax		ST0 empty 0.0
00403122	53	push ebx		ST1 empty 0.0
00403123	FF52 54	call dword ptr ds:[edx+0x54]		ST2 empty 0.0
00403126	85C0	test eax,ecx		

1-6 取最后一次循环的计算结果, 然后舍去小数部分-> username[usernameLength-1]*432.4*17.79/x0x15

地址	HEX 数据	反汇编	注释	寄存器 (FPU)
004032D6	. 50	push eax		EAX 00000000
004032D7	. FF15 34104000	call dword ptr ds:[<&MSVBVM60. __vbaHresul	msvbvm60. __vbaHresultCheckObj	ECX 868A3A8C
004032DD	> 8B55 D8	mov edx, dword ptr ss:[ebp-0x28]		EDX 004052F8 Co
004032E0	. 52	push edx		EBX 00000047
004032E1	. FF15 24104000	call dword ptr ds:[<&MSVBVM60. #516>]	[String = "t1捌01捌Y厘r叠 裂爆硝爆0 r? ? "	ESP 0012F2D4
004032E7	. 0FBFC0	movsx eax, ax		EBP 0012F49C
004032EA	. 8B4D D4	mov ecx, dword ptr ss:[ebp-0x2C]	eax=username[0]的ASCII值	ESI 0023C568 UN
004032ED	. 8985 60FEFFFF	mov edx, dword ptr ss:[ebp-0x1A0], eax	ecx=最后一次计算结果的整数部分	EDI 72A19FF1 ms
004032F3	. DB85 60FEFFFF	fld dword ptr ss:[ebp-0x1A0]	将username[0]的ASCII值转为浮点数	EIP 0040331F Co
004032F9	. 51	push ecx		C 0 ES 0023 32
004032FA	. DD9D 58FEFFFF	fstp qword ptr ss:[ebp-0x1A8]	保存username[0]的ASCII值到[ebp-0x1A8]	P 1 CS 001B 32
00403300	. FF15 B0104000	call dword ptr ds:[<&MSVBVM60. __vbaR8Str>	将最后一次计算结果的整数部分转为浮点数	A 0 SS 0023 32
00403306	. DC85 58FEFFFF	fadd qword ptr ss:[ebp-0x1A8]	最后一次计算结果的整数部分+username[0]的ASCII值	Z 1 DS 0023 32
0040330C	. 8B16	mov edx, dword ptr ds:[esi]	Colormas.004052F8	S 0 FS 003B 32
0040330E	. 56	push esi		T 0 GS 0000 NUI
0040330F	. C785 0CFEFFFF 05	mov dword ptr ss:[ebp-0xF4], 0x5		D 0
00403319	. DD9D 14FEFFFF	fstp qword ptr ss:[ebp-0xEC]	[ebp-0xEC]=最后一次计算结果的整数部分+username[0]	O 0 LastErr ERI
0040331F	. DFE0	fstsw ax		EFL 00000246 (N
00403321	. A8 0D	test al, 0xD		ST0 empty 0.0
00403323	. 0F85 52050000	jnz Colormas.0040387B		ST1 empty 0.0
00403329	. FF92 20030000	call dword ptr ds:[edx+0x320]	msvbvm60. 72A442F8	ST2 empty 0.0
0040332F	. 50	push eax		ST3 empty 0.0
Stack ss:[0012F380]-56994.000000000000				
地址	64 位双精度			地址 数值
0012F3B0	56994.000000000000	9.881312916824928e-324		0012F2D4 0023C
0012F3C0	2.635739459941128e-308	3.197242004116333e-241		0012F2D8 0012F
0012F3D0	0.0	1.018557979663330e-312		0012F2DC 0012F
0012F3E0	1.101167106045945e-306	2.121995790965272e-314		0012F2E0 000000
0012F3F0	-NAN FFFFFFFF 00002015	1.913430467494246e+257		0012F2E4 75A7B
0012F400	1.160808178569392e-315	1.101167106045946e-306		0012F2E8 000000
0012F410	56923.73039999999	0.0		0012F2EC 75A28
0012F420	1.903598048269742e+185	2.636087467250846e-308		0012F2F0 000000

1-7 将1-6的结果转为浮点数后加上用户名的第一个字节的ASCII值 这个就是完整的第一部分的算法了

用代码表示这部分的算法结果如下：

```
(username[usernameLength - 1] * 432.4*17.79 / 15 + username[0])
```

剩下的几部分算法也都是基于这个部分的结果

第二部分

地址	HEX 数据	反汇编	注释	寄存器 (FPU)
00403343	. FF51 50	call dword ptr ds:[ecx+0x50]		EAX 00000000
00403346	. 85C0	test eax, eax		ECX 868A3A8C
00403348	. DBE2	fclex		EDX 0012F214
0040334A	. 7D 15	jge short Colormas.00403361		EBX 00000047
0040334C	. 8B8D CCFEFFFF	mov ecx, dword ptr ss:[ebp-0x134]		ESP 0012F2D8
00403352	. 6A 50	push 0x50		EBP 0012F49C
00403354	. 68 A41E4000	push Colormas.00401EA4		ESI 0023C568 UNICODE "劲@"
00403359	. 51	push ecx		EDI 72A19FF1 msvbvm60. __vbaObj
0040335A	. 50	push eax		EIP 0040336B Colormas.0040336B
0040335B	. FF15 34104000	call dword ptr ds:[<&MSVBVM60. __vbaHresultCh	msvbvm60. __vbaHresultCheckObj	C 0 ES 0023 32位 0 (FFFFFFFF)
00403361	> 8B55 D0	mov edx, dword ptr ss:[ebp-0x30]		P 1 CS 001B 32位 0 (FFFFFFFF)
00403364	. 52	push edx		A 0 SS 0023 32位 0 (FFFFFFFF)
00403365	. FF15 B0104000	call dword ptr ds:[<&MSVBVM60. __vbaR8Str>]	将第一部分的计算结果转为浮点数	Z 1 DS 0023 32位 0 (FFFFFFFF)
0040336B	. 66:6DBE 19	imul bx, bx, 0x19	bx=username[0]*0x19	S 0 FS 003B 32位 7FFDF000 (FFF)
0040336F	. 0F80 0B050000	jg Colormas.00403880	结果不能溢出	T 0 GS 0000 NULL
00403375	. 0FBFC3	movsx eax, bx		D 0
00403378	. 8985 54FEFFFF	mov dword ptr ss:[ebp-0x1AC], eax	eax=username[0]*0x19	O 0 LastErr ERROR_SUCCESS (00C
0040337E	. 8D4D 8C	lea ecx, dword ptr ss:[ebp-0x74]	[ebp-0x1AC]==username[0]*0x19	EFL 00000246 (NO, NB, E, BE, NS, PE,
00403381	. DB85 54FEFFFF	fld dword ptr ss:[ebp-0x1AC]	将[ebp-0x1AC]转为浮点数	ST0 valid 56923.0000000000000000
00403387	. 8D95 7CFEFFFF	lea edx, dword ptr ss:[ebp-0x84]	edx="15"	ST1 empty 0.0
0040338D	. 51	push ecx		ST2 empty 0.0
0040338E	. 52	push edx		

2-1 首先将用户名的第一位的ASCII值乘以0x19->username[0]*0x19

地址	HEX 数据	反汇编	注释	寄存器
0040335A	. 50	push eax		EAX 000
0040335B	. FF15 34104000	call dword ptr ds:[<&MSVBVM60. __vbaHResultC	msvbvm60. __vbaHResultCheckObj	ECX 001
00403361	> 8B55 D0	mov edx, dword ptr ss:[ebp-0x30]		EDX 001
00403364	. 52	push edx		EBX 000
00403365	. FF15 B0104000	call dword ptr ds:[<&MSVBVM60. __vbaR8Str>]	将第一部分的计算结果转为浮点数	ESP 001
0040336B	. 66:6BDB 19	imul bx, bx, 0x19	bx=username[0]*0x19	EBP 001
0040336F	. 0F80 0B050000	ja Colormas.00403880	结果不能溢出	ESI 002
00403375	. 0FBFC3	movsx eax, bx	eax=username[0]*0x19	EDI 72A
00403378	. 8985 54FEFFFF	mov dword ptr ss:[ebp-0x1AC], eax	[ebp-0x1AC]==username[0]*0x19	EIP 004
0040337E	. 8D4D 8C	lea ecx, dword ptr ss:[ebp-0x74]		C 0 ES
00403381	. DB85 54FEFFFF	fild dword ptr ss:[ebp-0x1AC]	将[ebp-0x1AC]转为浮点数	P 0 CS
00403387	. 8D95 7CFFFFFF	lea edx, dword ptr ss:[ebp-0x84]	edx="15"	A 0 SS
0040338D	. 51	push ecx		Z 0 DS
0040338E	. 52	push edx		S 0 FS
0040338F	. C745 8C 05000000	mov dword ptr ss:[ebp-0x74], 0x5		T 0 GS
00403396	. DD9D 4CFEFFFF	fstp qword ptr ss:[ebp-0x1B4]	[ebp-0x1B4]=username[0]*0x19	D 0
0040339C	. DCA5 4CFEFFFF	fsub qword ptr ss:[ebp-0x1B4]	最后一次计算的结果-(username[0]*0x19)	0 0 La
004033A2	. DD5D 94	fstp qword ptr ss:[ebp-0x6C]	[ebp-0x6C]=最后一次计算的结果-(username[0]*0x19)	EFL 000
004033A5	. DFE0	fstsw ax		ST0 emp
004033A7	. A8 0D	test al, 0xD		ST1 emp
004033A9	. 0F85 CC040000	jnz Colormas.0040387B	第二部分	ST2 emp
004033AF	. FF15 C0104000	call dword ptr ds:[<&MSVBVM60. #573>]	msvbvm60. rtcHexVarFromVar	ST3 emp
ax=06EF				
地址	64 位双精度			地址
0012F430	55148.000000000000	1.397887401828547e-316		0012F2D0
0012F440	0.0	1.480761380687919e-300		0012F2D4
0012F450	1.479884167570145e-300	0.0		0012F2D8
0012F460	0.0	5.556318661224958e-308		0012F2DC
0012F470	5.556658180525371e-308	2.634754853894122e-308		0012F2E0
0012F480	3.952525166729972e-323	1.787783151976413e-307		0012F2E4
0012F490	1.787460608615478e-307	2.636112931200338e-308		0012F2E8

2-2 用1-4的结果减去2-1的结果，->(username[usernameLength - 1] * 432.4*17.79 / 15 - username[0] * 0x19)

第三部分

第三部分直接就是第一部分以十六进制形式转成字串的结果->username[usernameLength - 1] * 432.4*17.79 / 15

第四部分

地址	HEX 数据	反汇编	注释	寄存器 (FPU)
00403470	. 68 A0000000	push 0xA0		EAX 00000007
00403475	. 68 941E4000	push Colormas.00401E94		ECX 00401E94
0040347A	. 53	push ebx		EDX 0023EAC6 UNICODE "uiShou"
0040347B	. 50	push eax		EBX 00000047
0040347C	. FF15 34104000	call dword ptr ds:[<&MSVBVM60. __vbaHResultC	msvbvm60. __vbaHResultCheckObj	ESP 0012F2D8
00403482	> 8B55 C4	mov edx, dword ptr ss:[ebp-0x3C]		EBP 0012F49C
00403485	. 52	push edx		ESI 0023C568 UNICODE "为"
00403486	. FF15 24104000	call dword ptr ds:[<&MSVBVM60. #516>]	String = "u"	EDI 72A19FF1 msvbvm60. __vbaObjSet
0040348C	. 0FBFD8	movsx ebx, ax	rtcAnsiValueBstr	EIP 00403499 Colormas.00403499
0040348F	. 8B45 C0	mov eax, dword ptr ss:[ebp-0x40]	ebx=username[0]的ASCII值	C 0 ES 0023 32位 0(FFFFFFFF)
00403492	. 50	push eax		P 0 CS 001B 32位 0(FFFFFFFF)
00403493	. FF15 10104000	call dword ptr ds:[<&MSVBVM60. __vbaLenBstr>]	String = 00000007 ???	A 0 SS 0023 32位 0(FFFFFFFF)
00403499	. 0FAFD8	imul ebx, eax	_vbaLenBstr	Z 0 DS 0023 32位 0(FFFFFFFF)
0040349C	. 8B8D ACEFFFFF	mov ecx, dword ptr ss:[ebp-0x154]	ebx=username[0]的ASCII值*strlen(username)	S 0 FS 003B 32位 7FFDF000(FFF)
004034A2	. C785 FCFEFFFF 03	mov dword ptr ss:[ebp-0x104], 0x3		T 0 GS 0000 NULL
004034AC	. 0F80 CE030000	ja Colormas.00403880		D 0
004034B2	. 83EB 1B	sub ebx, 0x1B	ebx=username[0]的ASCII值*strlen(username)-0x1B	0 0 LastErr ERROR_SUCCESS (00000000)
004034B5	. 8D95 0CFEFFFF	lea edx, dword ptr ss:[ebp-0xF4]		EFL 00000202 (NO, NB, NE, A, NS, PO, GE, G)
004034BB	. 0F80 BF030000	ja Colormas.00403880		ST0 empty 0.0
004034C1	. 899D 04FEFFFF	mov dword ptr ss:[ebp-0xFC], ebx		ST1 empty 0.0
004034C7	. 8B19	mov ebx, dword ptr ds:[ecx]		ST2 empty 0.0
004034C9	. 8D85 7CFEFFFF	lea eax, dword ptr ss:[ebp-0x84]	eax="D76C" ? ? ?	ST3 empty 0.0
ebx=00000007				
地址	64 位双精度			地址
0012F430	55148.000000000000	1.479884167570063e-300		0012F2D8
0012F440	1.480761380687919e-300	1.480761380687919e-300		0012F2E8

第四部分就是用username[0]*用户名的长度再减去0x1B->username[0] * usernameLength - 0x1b

第五部分

地址	HEX	数据	反汇编	注释	寄存器 (FPU)
0040364A	68	A0000000	push 0xA0		EAX 002392F4 UNICODE "7"
0040364F	68	941E4000	push Colormas.00401E94		ECX 0012F468
00403654	56		push esi	msvbvm60.__vbaStrMove	EDX 002392F4 UNICODE "7"
00403655	50		push eax		EBX 01AFB9B4
00403656	FF15	34104000	call dword ptr ds:[<MSVBVM60.__vbaHresultCheckObj>]	msvbvm60.__vbaHresultCheckObj	ESP 0012F2D0
0040365C	>	8B45 D8	mov eax,dword ptr ss:[ebp-0x28]		EBP 0012F49C
0040365F	8B4D	D4	mov ecx,dword ptr ss:[ebp-0x2C]		ESI 72A26C30 msvbvm60.__vbaStrMove
00403662	8B55	D0	mov edx,dword ptr ss:[ebp-0x30]		EDI 72A19FF1 msvbvm60.__vbaObjSet
00403665	50		push eax		EIP 00403682 Colormas.00403682
00403666	51		push ecx		C 0 ES 0023 32位 0(FFFFFFFF)
00403667	52		push edx		P 1 CS 001B 32位 0(FFFFFFFF)
00403668	FF15	10104000	call dword ptr ds:[<MSVBVM60.__vbaLenBstr>]	[String = "7" 求用户名长度	A 0 SS 0023 32位 0(FFFFFFFF)
0040366E	50		push eax		Z 1 DS 0023 32位 0(FFFFFFFF)
0040366F	FF15	08104000	call dword ptr ds:[<MSVBVM60.__vbaStrI4>]	将用户名长度转为字符串	S 0 FS 003B 32位 7FFDF000 (FFF)
00403675	8B35	DC104000	mov esi,dword ptr ds:[<MSVBVM60.__vbaStrMove>]	msvbvm60.__vbaStrMove	T 0 GS 0000 NULL
0040367B	8BD0		mov edx,eax		D 0
0040367D	8D4D	CC	lea ecx,dword ptr ss:[ebp-0x34]		0 0 LastErr ERROR_SUCCESS (00000000)
00403680	FFD6		call esi	将序列号保存到[ebp-0x34]: <MSVBVM60.__vbaStrMove>	EFL 00000246 (NO,NB,E, BE, NS, PE, GE, LE
00403682	8B3D	30104000	mov edi,dword ptr ds:[<MSVBVM60.__vbaStrCat>]	msvbvm60.__vbaStrCat	ST0 empty 0.0
00403688	50		push eax	[String = "7"	ST1 empty 0.0
00403689	FFD7		call edi	将序列号和用户名长度进行拼接 ->56994 D76C DE5B 470	ST2 empty 0.0
0040368B	8BD0		mov edx,eax		ST3 empty 0.0
ds:[00401834]-72A1A274 (msvbvm60.__vbaHresultCheckObj)					

第五部分就是用户名的长度

第六部分

地址	HEX	数据	反汇编	注释	寄存器 (FPU)	
00403675	8B35	DC104000	mov esi,dword ptr ds:[<MSVBVM60.__vbaStrMove	msvbvm60.__vbaStrMove	EAX 0023F86C UNICODE "56994D76CDE5B4707"	
0040367B	8BD0		mov edx,eax		ECX 0012F464	
0040367D	8D4D	CC	lea ecx,dword ptr ss:[ebp-0x34]		EDX 0023F86C UNICODE "56994D76CDE5B4707"	
00403680	FFD6		call esi	将序列号保存到[ebp-0x34]: <MSVBVM60.__vbaStrMove	EBX 01AFB9B4	
00403682	8B3D	30104000	mov edi,dword ptr ds:[<MSVBVM60.__vbaStrCat>	msvbvm60.__vbaStrCat	ESP 0012F2C0	
00403688	50		push eax	[String = "5"	EBP 0012F49C	
00403689	FFD7		call edi	将序列号和用户名长度进行拼接 ->56994 D76C DE5B 470	ESI 72A26C30 msvbvm60.__vbaStrMove	
0040368B	8BD0		mov edx,eax		EDI 72A26A76 msvbvm60.__vbaStrCat	
0040368D	8D4D	C8	lea ecx,dword ptr ss:[ebp-0x38]		EIP 00403698 Colormas.00403698	
00403690	FFD6		call esi	msvbvm60.__vbaStrMove	C 0 ES 0023 32位 0(FFFFFFFF)	
00403692	50		push eax		P 1 CS 001B 32位 0(FFFFFFFF)	
00403693	68	741F4000	push Colormas.00401F74	-CM	A 0 SS 0023 32位 0(FFFFFFFF)	
00403698	FFD7		call edi	将序列号和CM进行拼接->56994D76CDE5B4707-CM	Z 1 DS 0023 32位 0(FFFFFFFF)	
0040369A	8BD0		mov edx,eax		S 0 FS 003B 32位 7FFDF000 (FFF)	
0040369C	8D4D	C4	lea ecx,dword ptr ss:[ebp-0x3C]		T 0 GS 0000 NULL	
0040369F	FFD6		call esi	将序列号保存到[ebp-0x34]	D 0	
004036A1	50		push eax		0 0 LastErr ERROR_SUCCESS (00000000)	
004036A2	FF15	74104000	call dword ptr ds:[<MSVBVM60.__vbaStrCat>	比较计算的序列号和输入的序列号	EFL 00000246 (NO,NB,E, BE, NS, PE, GE, LE)	
004036A8	8BF0		mov esi,eax		ST0 empty 0.0	
004036AA	8D45	C4	lea eax,dword ptr ss:[ebp-0x3C]		ST1 empty 0.0	
004036AD	8D4D	D8	lea ecx,dword ptr ss:[ebp-0x28]		ST2 empty 0.0	
004036B0	50		push eax		ST3 empty 0.0	
edi-72A26A76 (msvbvm60.__vbaStrCat)						
地址	64 位双精度			地址	数值	注释
0012F430	55148.000000000000	0.0		0012F2CC	00401F74	UNICODE "-CM"
0012F440	0.0	1.479884162879531e-300		0012F2D0	0023F86C	UNICODE "56994D76CDE5B4707"
0012F450	1.479536700036318e-300	0.0		0012F2D4	0023FA9C	UNICODE "123456789"
0012F460	5.554451302599057e-308	5.556658180525371e-308		0012F2D8	0012F4A8	
0012F470	5.556827940205446e-308	2.634754853894122e-308		0012F2DC	0012F578	UNICODE "I"
0012F480	3.952525166729972e-323	1.787783151976413e-307		0012F2E0	00000001	
0012F490	1.787460608615478e-307	2.636112931200338e-308		0012F2E4	75A7B01C	usp10.75A7B01C
0012F4A0	5.499026672356932e-308	1.798338807838472e-307		0012F2E8	00000000	
0012F4B0	1.797231975269503e-307	1.072250582006232e+244		0012F2EC	409BBC00	
0012F4C0	2.636545820419276e-308	9.881312916824928e-324		0012F2F0	0000006F	

第六部分是固定的字符串"-CM"，拼接完第六部分之后就算关键的比较函数了。

以上就是这个程序的完整的算法，另外这个作者好像还有点皮，会用SendKey来模拟键盘按键给你捣乱，但对调试程序影响不大。

写出注册机

根据每一部分的算法 我们可以写出这个程序的注册机 代码如下：

```
#include <iostream>
#include <windows.h>

using namespace std;

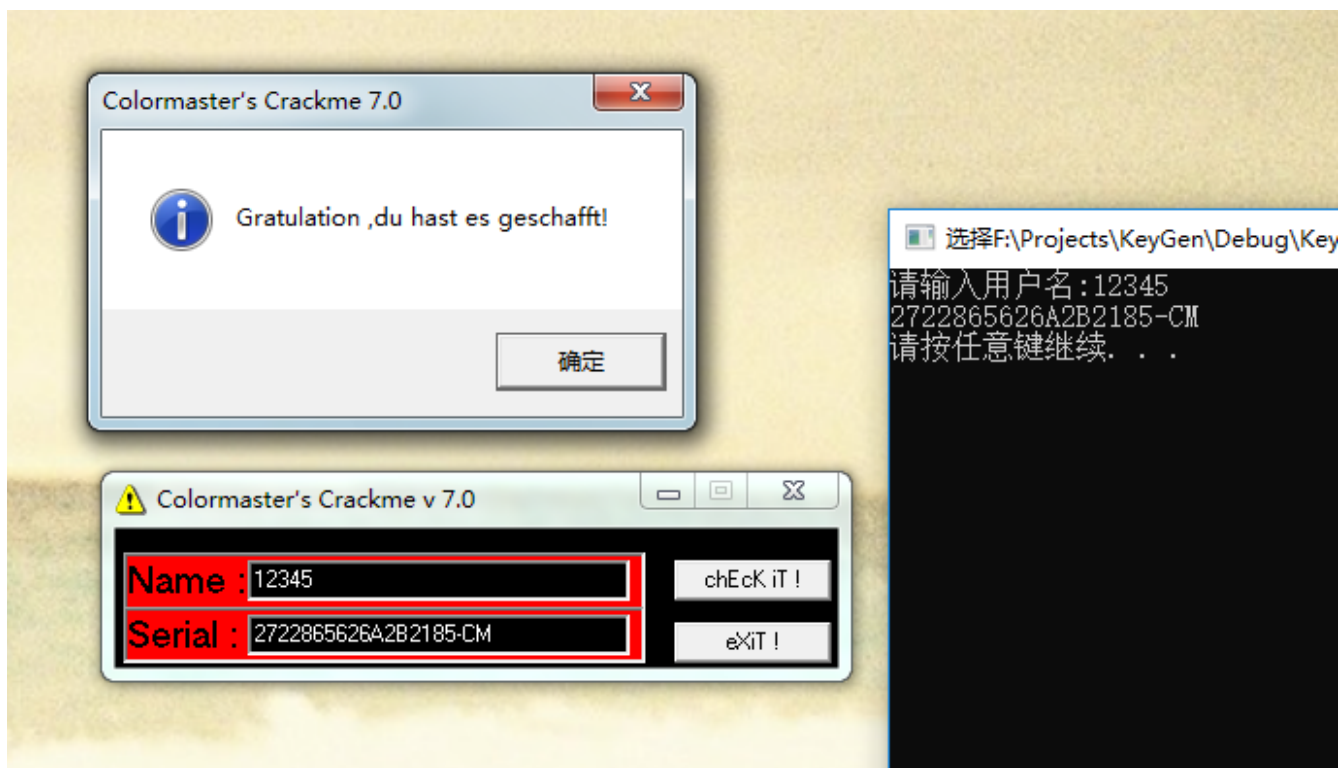
int main()
{
    char username[20] = { 0 };
    printf("请输入用户名:");
```

```

scanf_s("%s", username, 20);
int usernameLength = strlen(username);
if (usernameLength < 5)
{
    printf("用户名长度必须大于5");
}
char key[50];
char key1[10], key2[10], key3[10], key4[10], key5[10];
sprintf(key1, "%d", (int)(username[usernameLength - 1] * 432.4*17.79 / 15 +
username[0]));
sprintf(key2, "%x", (int)(username[usernameLength - 1] * 432.4*17.79 / 15 -
username[0] * 0x19));
sprintf(key3, "%x", (int)(username[usernameLength - 1] * 432.4*17.79 / 15));
sprintf(key4, "%d", (int)(username[0] * usernameLength - 0x1b));
sprintf(key5, "%d", usernameLength);
memset(key, 0, sizeof(key));
strcat(key, key1);
strcat(key, key2);
strcat(key, key3);
strcat(key, key4);
strcat(key, key5);
strcat(key, "-CM");
printf("%s\n", key);
system("pause");
return 0;
}

```

验证结果



输入注册机算出的序列号，提示成功 破解完成

最后，需要udd相关文件的可以到我的Github下载: <https://github.com/TonyChen56/160-Crackme>