
Subject: Re: Protect package - A starting copy protection system

Posted by [Tom1](#) on Wed, 06 Jun 2018 10:16:08 GMT

[View Forum Message](#) <> [Reply to Message](#)

Max,

32-bit is now OK on all my compilers (MSVS15, MSVS17, MSBT17).

However, regardless of compiler, 64-bit executable crashes now immediately. Here's the MSVS17x64 ProtectEncrypt log on TheIDE compiler console:

Linking...

ProtectTest.exe

ENCRYPTION KEY : aabbccddeeff00112233445566778899

```
LEN:5  e8 b4 26 03 00          call 0x326b9
LEN:3  8d 04 1b                lea eax, ptr [ebx+ebx*1]
LEN:4  89 44 24 50             mov dword ptr [esp+0x50], eax
LEN:5  e8 08 2d 00 00          call 0x2d0d
LEN:1  48                      dec eax
LEN:2  8b d8                   mov ebx, eax
LEN:1  48                      dec eax
LEN:6  8d 15 1e 4f 17 00        lea edx, ptr [0x174f1e]
LEN:1  48                      dec eax
LEN:2  8b c8                   mov ecx, eax
LEN:5  e8 e6 41 00 00          call 0x41eb
LEN:1  48                      dec eax
LEN:4  8d 54 24 50             lea edx, ptr [esp+0x50]
LEN:1  48                      dec eax
LEN:4  8d 4c 24 28             lea ecx, ptr [esp+0x28]
LEN:5  e8 37 fb ff ff          call 0xfffffb3c
LEN:1  48                      dec eax
LEN:2  8b d0                   mov edx, eax
LEN:4  0f b6 40 0e             movzx eax, byte ptr [eax+0xe]
LEN:2  84 c0                   test al, al
LEN:2  75 07                   jnz 0x9
LEN:1  44                      inc esp
LEN:4  0f be 42 0f             movsx eax, byte ptr [edx+0xf]
LEN:2  eb 04                   jmp 0x6
LEN:1  44                      inc esp
LEN:3  8b 42 08                mov eax, dword ptr [edx+0x8]
LEN:2  84 c0                   test al, al
LEN:2  74 03                   jz 0x5
LEN:1  48                      dec eax
LEN:2  8b 12                   mov edx, dword ptr [edx]
LEN:1  49                      dec ecx
LEN:2  63 f8                   arpl ax, di
LEN:1  48                      dec eax
LEN:3  8b 4b 18                mov ecx, dword ptr [ebx+0x18]
```

LEN:1 48	dec eax
LEN:3 8d 04 39	lea eax, ptr [ecx+edi*1]
LEN:1 48	dec eax
LEN:3 3b 43 28	cmp eax, dword ptr [ebx+0x28]
LEN:2 77 0e	jnb 0x10
LEN:1 4c	dec esp
LEN:2 8b c7	mov eax, edi
LEN:5 e8 11 76 15 00	call 0x157616
LEN:1 48	dec eax
LEN:3 01 7b 18	add dword ptr [ebx+0x18], edi
LEN:2 eb 09	jmp 0xb
LEN:1 48	dec eax
LEN:2 8b 03	mov eax, dword ptr [ebx]
LEN:1 48	dec eax
LEN:2 8b cb	mov ecx, ebx
LEN:2 ff 10	call dword ptr [eax]
LEN:1 90	nop
LEN:5 80 7c 24 36 00	cmp byte ptr [esp+0x36], 0x0
LEN:2 74 0b	jz 0xd
LEN:1 48	dec eax
LEN:4 8d 4c 24 28	lea ecx, ptr [esp+0x28]
LEN:5 e8 61 14 00 00	call 0x1466
LEN:1 90	nop
LEN:1 48	dec eax
LEN:6 8d 15 a5 4e 17 00	lea edx, ptr [0x174ea5]
LEN:1 48	dec eax
LEN:2 8b cb	mov ecx, ebx
LEN:5 e8 71 41 00 00	call 0x4176
LEN:5 e8 89 25 03 00	call 0x3258e
LEN:5 e8 e4 2b 00 00	call 0x2be9
LEN:1 48	dec eax
LEN:2 8b f8	mov edi, eax
LEN:1 48	dec eax
LEN:6 8d 15 0a 4e 17 00	lea edx, ptr [0x174e0a]
LEN:1 48	dec eax
LEN:2 8b c8	mov ecx, eax
LEN:5 e8 c2 40 00 00	call 0x40c7
LEN:4 0f b6 43 0e	movzx eax, byte ptr [ebx+0xe]
LEN:2 84 c0	test al, al
LEN:2 75 07	jnz 0x9
LEN:1 44	inc esp
LEN:4 0f be 43 0f	movsx eax, byte ptr [ebx+0xf]
LEN:2 eb 04	jmp 0x6
LEN:1 44	inc esp
LEN:3 8b 43 08	mov eax, dword ptr [ebx+0x8]
LEN:2 84 c0	test al, al
LEN:2 74 03	jz 0x5
LEN:1 48	dec eax

LEN:2	8b 1b	mov ebx, dword ptr [ebx]
LEN:1	49	dec ecx
LEN:2	63 f0	arpl ax, si
LEN:1	48	dec eax
LEN:3	8b 4f 18	mov ecx, dword ptr [edi+0x18]
LEN:1	48	dec eax
LEN:3	8d 04 31	lea eax, ptr [ecx+esi*1]
LEN:1	48	dec eax
LEN:2	8b d3	mov edx, ebx
LEN:1	48	dec eax
LEN:3	3b 47 28	cmp eax, dword ptr [edi+0x28]
LEN:2	77 0e	jnb 0x10
LEN:1	4c	dec esp
LEN:2	8b c6	mov eax, esi
LEN:5	e8 fc 74 15 00	call 0x157501
LEN:1	48	dec eax
LEN:3	01 77 18	add dword ptr [edi+0x18], esi
LEN:2	eb 08	jmp 0xa
LEN:1	48	dec eax
LEN:2	8b 07	mov eax, dword ptr [edi]
LEN:1	48	dec eax
LEN:2	8b cf	mov ecx, edi
LEN:2	ff 10	call dword ptr [eax]
LEN:1	48	dec eax
LEN:6	8d 15 a3 4d 17 00	lea edx, ptr [0x174da3]
LEN:1	48	dec eax
LEN:2	8b cf	mov ecx, edi
LEN:5	e8 6f 40 00 00	call 0x4074

ENCRYPT RESULTS:

Code sequences : 1

Data sequences : 2

Obfuscate sequences : 1

C:\upp-11979\out\MyApps\MSVS17x64.Protect\ProtectTest.exe (3476480 B) linked in (0:03.04)

OK. (0:34.67)

The ProtectTest.log looks like this:

* C:\upp-11979\out\MyApps\MSVS17x64.Protect\ProtectTest.exe 06.06.2018 13:06:22, user: tom

START DECRYPT

LEN: 5 - e8 b4 26 03 00	call 0x326b9
LEN: 3 - 8d 04 1b	lea eax, ptr [rbx+rbx*1]
LEN: 4 - 89 44 24 50	mov dword ptr [rsp+0x50], eax
LEN: 5 - e8 08 2d 00 00	call 0x2d0d
LEN: 3 - 48 30 d8	xor al, bl
LEN: 1 - 5e	pop rsi
LEN: 1 - 50	push rax

LEN: 2 - 65 1e	invalid
LEN: 1 - 6c	insb
LEN: 1 - a5	movsd dword ptr [rdi], dword ptr [rsi]
LEN: 3 - 0a 6b 39	or ch, byte ptr [rbx+0x39]
LEN: 1 - 2f	invalid
LEN: 1 - fc	cld
LEN: 1 - 54	push rsp
LEN: 5 - b8 00 00 af 93	mov eax, 0x93af0000
LEN: 5 - 2d 24 50 fa 74	sub eax, 0x74fa5024
LEN: 2 - b0 24	mov al, 0x24
LEN: 5 - bd 91 37 fb ff	mov ebp, 0xffffb3791
LEN: 2 - 70 b4	jo 0xffffffffffffb6
LEN: 2 - 3b d0	cmp edx, eax
LEN: 1 - 56	push rsi
LEN: 3 - 49 40 0e	invalid
LEN: 1 - 5f	pop rdi
LEN: 6 - 12 a3 07 bb f1 be	adc ah, byte ptr [rbx-0x410e44f9]
LEN: 2 - 8a 0f	mov cl, byte ptr [rdi]
LEN: 2 - cd 04	int 0x4
LEN: 3 - f2 7f 42	bnd jnle 0x45
LEN: 2 - 13 fa	adc edi, edx
LEN: 1 - f4	hlt
LEN: 5 - 15 03 7c f1 12	adc eax, 0x12f17c03
LEN: 2 - 49 ce	invalid
LEN: 1 - 55	push rbp
LEN: 4 - 48 23 4b 18	and rcx, qword ptr [rbx+0x18]
LEN: 2 - 4e aa	stosb byte ptr [rdi]
LEN: 1 - aa	stosb byte ptr [rdi]
LEN: 1 - 1e	invalid
LEN: 2 - 48 f8	clc
LEN: 3 - 80 28 89	sub byte ptr [rax], 0x89
LEN: 3 - f0 34 a8	lock xor al, 0xa8
LEN: 5 - bf 51 11 76 15	mov edi, 0x15761151
LEN: 4 - 23 4c 80 7b	and ecx, dword ptr [rax+rax*4+0x7b]
LEN: 9 - a1 69 09 c6 2b 03 3a 92 cb	mov eax, dword ptr [0xcb923a032bc60969]
LEN: 1 - 6e	outsb
LEN: 1 - 91	xchg ecx, eax
LEN: 2 - 77 7f	jnb 0x81
LEN: 3 - f2 24 36	and al, 0x36
LEN: 9 - a0 b5 0b 87 40 4c 24 28 2d	mov al, byte ptr [0x2d28244c40870bb5]
LEN: 3 - 13 14 00	adc edx, dword ptr [rax+rax*1]
LEN: 3 - 19 65 f2	sbb dword ptr [rbp-0xe], esp
LEN: 6 - f3 15 a5 4e 17 00	adc eax, 0x174ea5
LEN: 2 - df 29	fild st, qword ptr [rcx]
LEN: 2 - 34 7d	xor al, 0x7d
LEN: 2 - b0 41	mov al, 0x41
LEN: 1 - cf	iretd
LEN: 2 - cd eb	int 0xeb

BTW: Should I use 64-bit ProtectEncrypt to process 64-bit executables or is it OK to use the same 32-bit version as for 32-bit executables? (Now I used 32-bit ProtectEncrypt for 64-bit executables, but that's the way I did before too.)

BR,

Tom

EDIT: Yes, answering my own question: ProtectEncrypt must be 64-bit version for 64-bit executables. Now it works beautifully!!! Congratulations Max! This is an excellent step forward!!! :)
