

纯手工秒杀 VM 虚拟机 Handle 之 II

/*

*/

/* 作者:半斤八兩

/* 博客:<http://cnblogs.com/bjblcracked>

/* 日期:2015-08-26

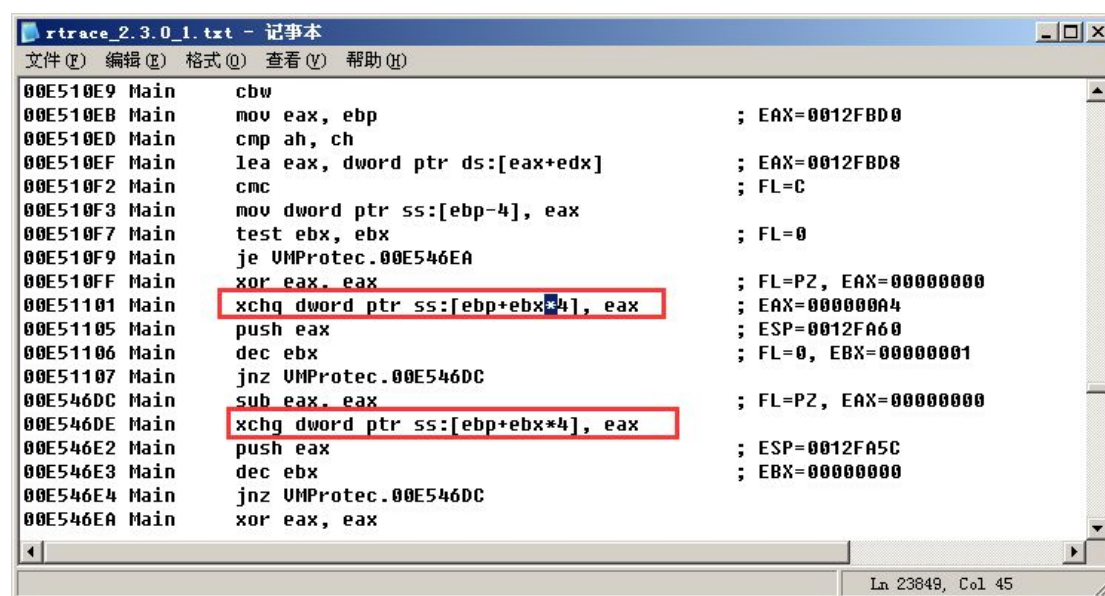
/*

*/

只是感兴趣，没有其他目的。失误之处敬请诸位大侠赐教!

在上一篇中，《纯手工秒杀 VM,SE 等虚拟机 Handle》我们介绍了快速定位 Handle 的方法，此法在 VMP 2.3.0 中已经失效了。在新版本中改动挺大的，导致之前许多前辈公布的工具都已经失效。我们先用老方法试试新版本。（过程，略。。。）不会的童鞋可参考上一篇

结果如下：



```
rtrace_2.3.0_1.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

00E510E9 Main    cbw
00E510EB Main    mov eax, ebp           ; EAX=0012FBD0
00E510ED Main    cmp ah, ch
00E510EF Main    lea eax, dword ptr ds:[eax+edx] ; EAX=0012FBD8
00E510F2 Main    cmc                   ; FL=C
00E510F3 Main    mov dword ptr ss:[ebp-4], eax
00E510F7 Main    test ebx, ebx         ; FL=0
00E510F9 Main    je VMProtec.00E546EA
00E510FF Main    xor eax, eax          ; FL=PZ, EAX=00000000
00E51101 Main    xchg dword ptr ss:[ebp+ebx*4], eax ; EAX=000000A4
00E51105 Main    push eax              ; ESP=0012FA60
00E51106 Main    dec ebx               ; FL=0, EBX=00000001
00E51107 Main    jnz VMProtec.00E546DC
00E546DC Main    sub eax, eax          ; FL=PZ, EAX=00000000
00E546DE Main    xchg dword ptr ss:[ebp+ebx*4], eax
00E546E2 Main    push eax              ; ESP=0012FA5C
00E546E3 Main    dec ebx               ; EBX=00000000
00E546E4 Main    jnz VMProtec.00E546DC
00E546EA Main    xor eax, eax

Ln 23849, Col 45
```

这次搜到的“*”符号结果和之前的版本完全不一样。

这次搜到的其实已经可以定位到关键地方了。但是这些信息对于我们新手来说好像根本看不出来有什么用？怎么办？试试新招。

我们想想 VMP 程序他有文件完整性效验，内存效验，断点检测，虚拟机检测，调试检测，等等。这些反调试手段我们都可以做为分析的切入点。我们就拿《文件完整性效验》来做演示。以下简称 VMP_CRC。

效验文件完整性一般有两种方式，效验磁盘上的 Bin 程序，效验内存中的 BIN 程序。而 VMP 2.3.0 主程序是把磁盘上的文件读到内存中再进行校验。VMP_CRC 喜欢分段校验。每校验一段的时候，肯定会有一个很大的循环。而这个大循环就是我们的切入点。Okay，知道这些，我们再来看一下主程序的大小。



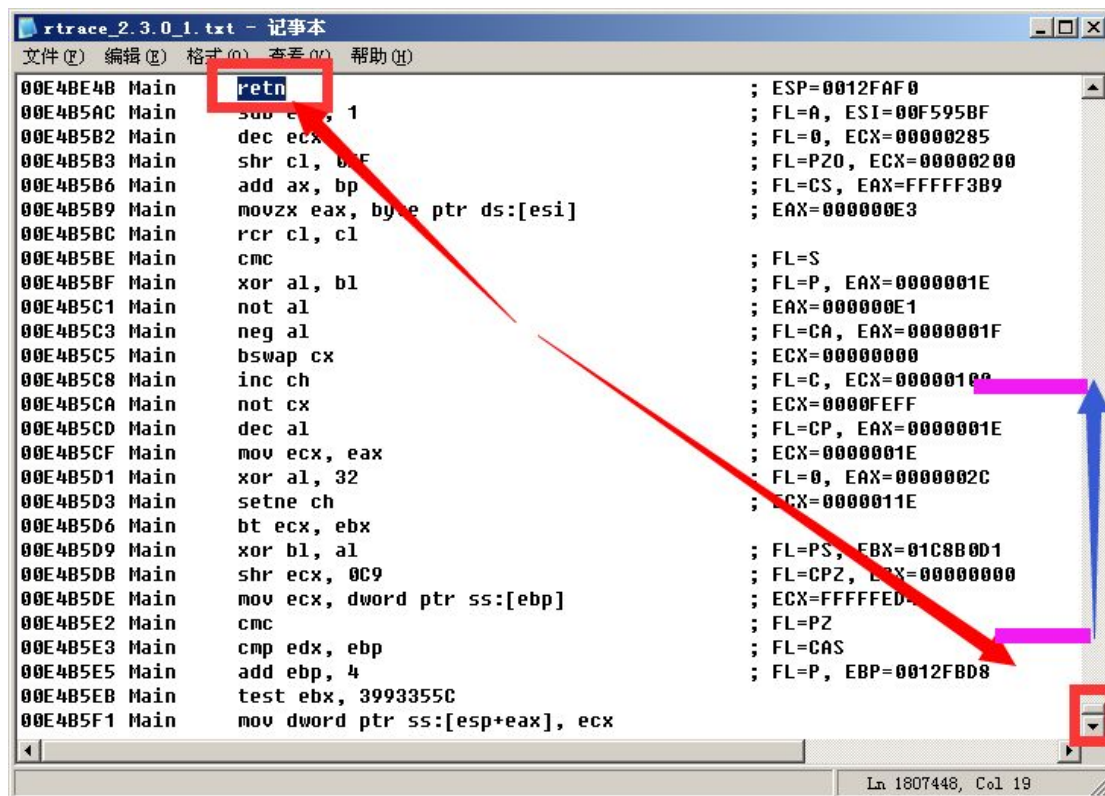
5.84 MB (6,125,592 字节) 天文数字

我们保守估计每条 `vm_crc` 只用了 5 条指令，那 $6125592 * 5$ 也有 30627960 条指令。仅仅是一个 `crc` 就有三千多万条指令需要执行。相信说到这里，大部分童鞋应该知道思路了吧？还不知道？没关系，我们来演示一下。

重复上一篇的步骤，只是这一次 `trace` 的时间由之前的 3 秒改成 3 分钟。然后用记事本打开这个庞大的 `trace` 数据。

打开之后，我们拉到记事本的底部，然后从下往上搜索 “`retn`” 指

令。搜到的结果如下。



```
00E4BE4B Main    retn                ; ESP=0012FAF0
00E4B54C Main    sub ecx, 1          ; FL=A, ESI=00F595BF
00E4B5B2 Main    dec ecx           ; FL=0, ECX=00000285
00E4B5B3 Main    shr cl, 0xF         ; FL=P20, ECX=00000200
00E4B5B6 Main    add ax, bp          ; FL=CS, EAX=FFFFFF3B9
00E4B5B9 Main    movzx eax, byte ptr ds:[esi] ; EAX=000000E3
00E4B5BC Main    rcr cl, cl
00E4B5BE Main    cmc
00E4B5BF Main    xor al, bl
00E4B5C1 Main    not al
00E4B5C3 Main    neg al
00E4B5C5 Main    bswap cx
00E4B5C8 Main    inc ch
00E4B5CA Main    not cx
00E4B5CD Main    dec al
00E4B5CF Main    mov ecx, eax
00E4B5D1 Main    xor al, 32
00E4B5D3 Main    setne ch
00E4B5D6 Main    bt ecx, ebx
00E4B5D9 Main    xor bl, al
00E4B5DB Main    shr ecx, 0C9
00E4B5DE Main    mov ecx, dword ptr ss:[ebp]
00E4B5E2 Main    cmc
00E4B5E3 Main    cmp edx, ebp
00E4B5E5 Main    add ebp, 4
00E4B5EB Main    test ebx, 3993355C
00E4B5F1 Main    mov dword ptr ss:[esp+eax], ecx ; FL=S
                                ; FL=P, EAX=0000001E
                                ; EAX=000000E1
                                ; FL=CA, EAX=0000001F
                                ; ECX=00000000
                                ; FL=C, ECX=00000100
                                ; ECX=0000FEFF
                                ; FL=CP, EAX=0000001E
                                ; ECX=0000001E
                                ; FL=0, EAX=0000002C
                                ; ECX=0000011E
                                ; FL=PS, EBX=01C8B0D1
                                ; FL=CPZ, ECX=00000000
                                ; ECX=FFFFFFED
                                ; FL=PZ
                                ; FL=CAS
                                ; FL=P, EBP=0012FB08
```

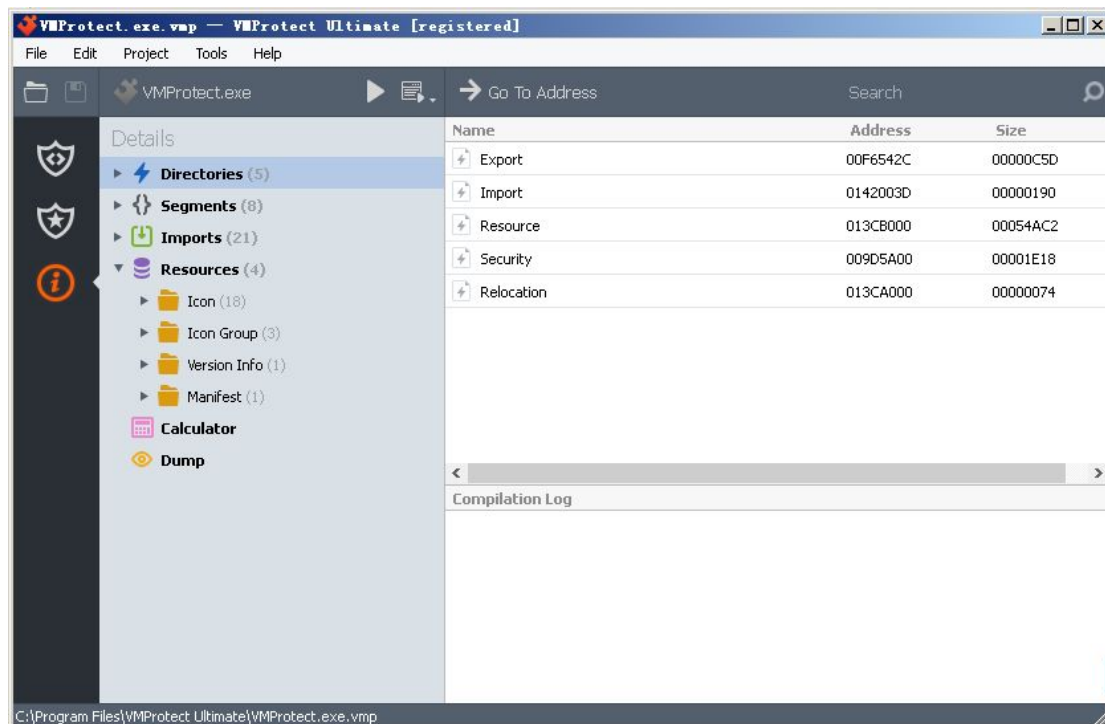
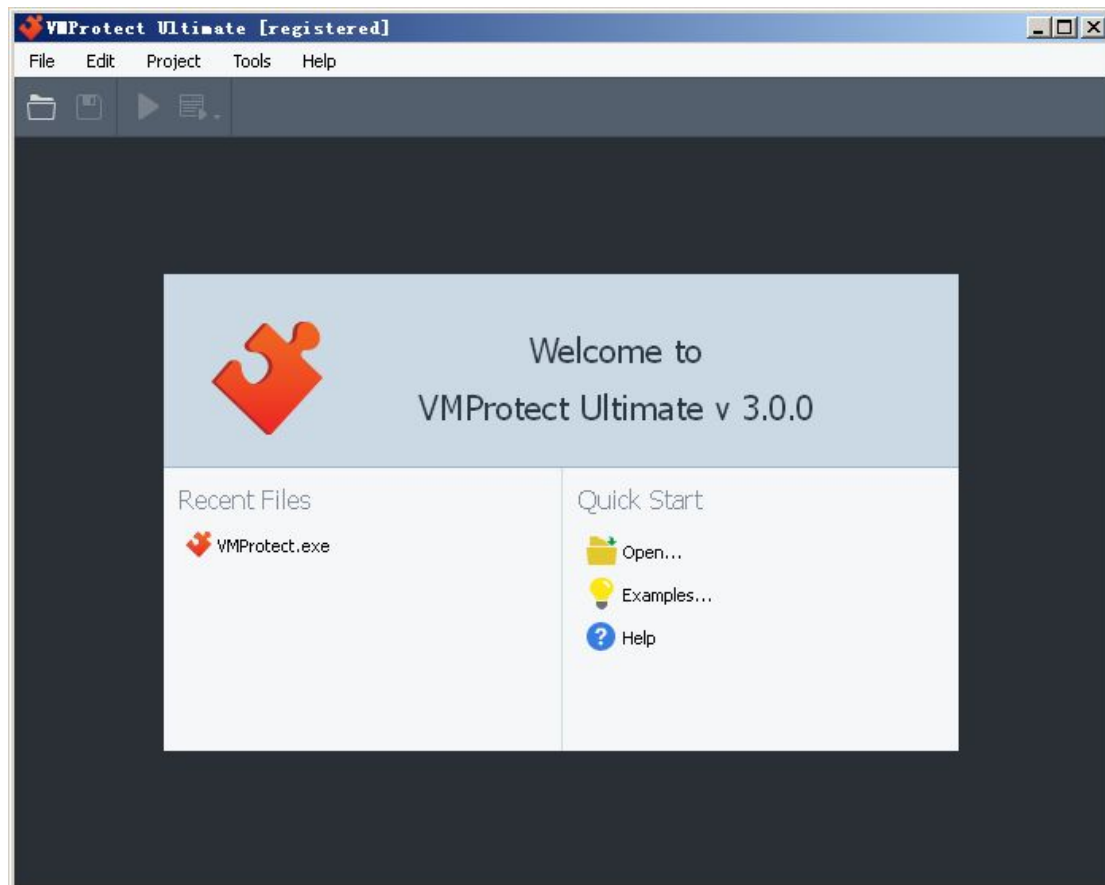
右下角状态栏可以看到我当前 trace 指令有近两百万条。现在我们不停的按 F3（查找上一个）。查找的同时注意看着滚动条。当滚动条跨度很大时，就说明这区间就是我们要找的 vm_crc 大循环了。如下图所示。

G.P.U* - main thread, module VMProtect			
Address	Hex dump	Disassembly	Comment
013BB5B7 <VMProtect.VM_CRC>	55	push ebp	VM_CRC
013BB5B8	0F31	rdtsc	
013BB5BA	99	cdq	
013BB5BB	8BEC	mov ebp, esp	
013BB5BD	66:0FA3D8	bt ax, bx	
013BB5C1	66:0FC8	bswap ax	
013BB5C4	8BC7	mov eax, edi	
013BB5C6	C6C6 25	mov dh, 25	
013BB5C9	81EA 1442B708	sub edx, 8B74214	
013BB5CF	99	cdq	
013BB5D0	F6C6 13	test dh, 13	
013BB5D3	83E2 03	and edx, 3	
013BB5D6	03C2	add eax, edx	
013BB5D8	0FA4F2 DE	shld edx, esi, 0DE	
013BB5DC	8AD2	mov dl, dl	
013BB5DE	8B55 08	mov edx, dword ptr ss:[ebp+8]	
013BB5E1	C1F8 02	sar eax, 2	
013BB5E4	F8	clc	
013BB5E5	3BD5	cmp edx, ebp	
013BB5E7	56	push esi	
013BB5E8	66:1BF0	sbb si, ax	
013BB5EB	0FB7F0	movzx esi, ax	
013BB5EE	0BF5	or esi, ebp	
013BB5F0	8D3482	lea esi, dword ptr ds:[edx*eax*4]	
013BB5F3	C0C6 7B	rol dh, 7B	
013BB5F6	D3F2	sal edx, cl	
013BB5F8	33C9	xor ecx, ecx	
013BB5FA	80FE 2C	cmp dh, 2C	
013BB5FD	F7D8	neg eax	
013BB5FF	0FB7D1	movzx edx, cx	
013BB602	66:99	cwd	
013BB604	8BD0	mov edx, eax	
013BB606	0F84 3B000000	je VMProtect.013BB647	
013BB60C	8D49 00	lea ecx, dword ptr ds:[ecx]	
013BB60F	66:0FBEC2	movsx ax, dl	
013BB613	8B0496	mov eax, dword ptr ds:[esi+edx*4]	
013BB616	69C0 512D9ECC	imul eax, eax, CC9E2D51	
013BB61C	F9	stc	
013BB61D	C1C0 0F	rol eax, 0F	
013BB620	69C0 9335871B	imul eax, eax, 1B873593	
013BB626	33C1	xor ecx, ecx	
013BB628	D3D9	rcr ecx, cl	
013BB62A	C1C0 0D	rol eax, 0D	
013BB62D	8BCE	mov ecx, esi	
013BB62F	F8	clc	
013BB630	83C2 01	add edx, 1	
013BB633	86C9	xchg cl, cl	
013BB635	B9 11303562	mov ecx, 62353011	
013BB63A	8D8C80 646B54E	lea ecx, dword ptr ds:[eax*eax*4+E6546B64]	
013BB641	0F85 CFFFFFFF	jnz VMProtect.013BB613	
013BB647	8BC7	mov eax, edi	
013BB649	83E0 03	and eax, 3	
013BB64C	F7C5 91620379	test ebp, 79036291	

找到了 vm_crc 就可以顺藤摸瓜把 vm_jump vm_pushxx vm_call 等。全部揪出来 :)

老毛子这次 VMP 更新变化很大。VMP 自身改动不小，就连 UI 都上 QT 了。目测老毛子下一步准备迈向移动市场了。。。

这次 2.3.0 版本的解压后大概 50 多 M，光 QT 库就占了几十 M -_-!!!



不得不承认，确实比以前漂亮许多。



另外此版本仅支持英语和老毛子的母语，不再支持中文。藐视我们这个神奇的国度已经伤透老毛子的心了。。。

半斤八兩[4st]

Bjblcracked[at]126.com

2015.08.26

感谢 bianfeng Kido 透明色

安装密码:aad50b3af89b9989af1d4592d7263817