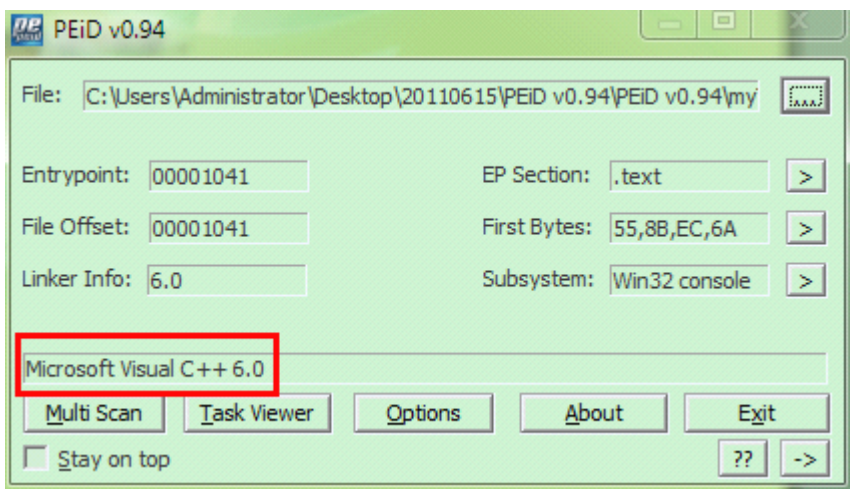
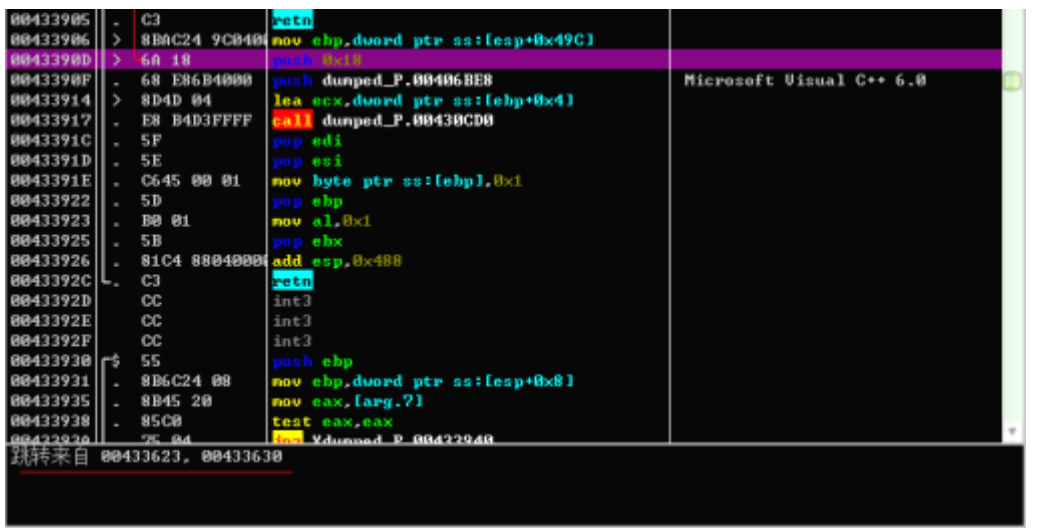
PEiD 0.94 简略分析

PEiD 是一款为大家所知的文件类型分析工具,，今天我就来分析一下 PEiD 识别的原理。我所用的版本是 PEid 0.94 版，它加了 PECompact 2.x -> Jeremy Collake 壳，在这里把壳脱掉便于分析，脱壳流程这里不在叙述。

我们把一个用 VC++6.0 生成的文件拖入 PEiD 进行观察.。



OD 载入 PEid，然后使用字符串搜索插件查找 Microsoft Visual C++ 6.0，找到引用的位置如下图:



从上图可知跳转来至两处判断后的流程转移，那我们就来分析一下 00433530 这个函数的实现，看需要什么条件才能使 PEiD 判断程序为 Microsoft Visual C++ 6.0 所生成的。关键代码如下:

```
00433530  /$  81EC 88040000 sub esp,0x488
00433536  |.  53           push ebx
00433537  |.  55           push ebp
00433538  |.  56           push esi
```

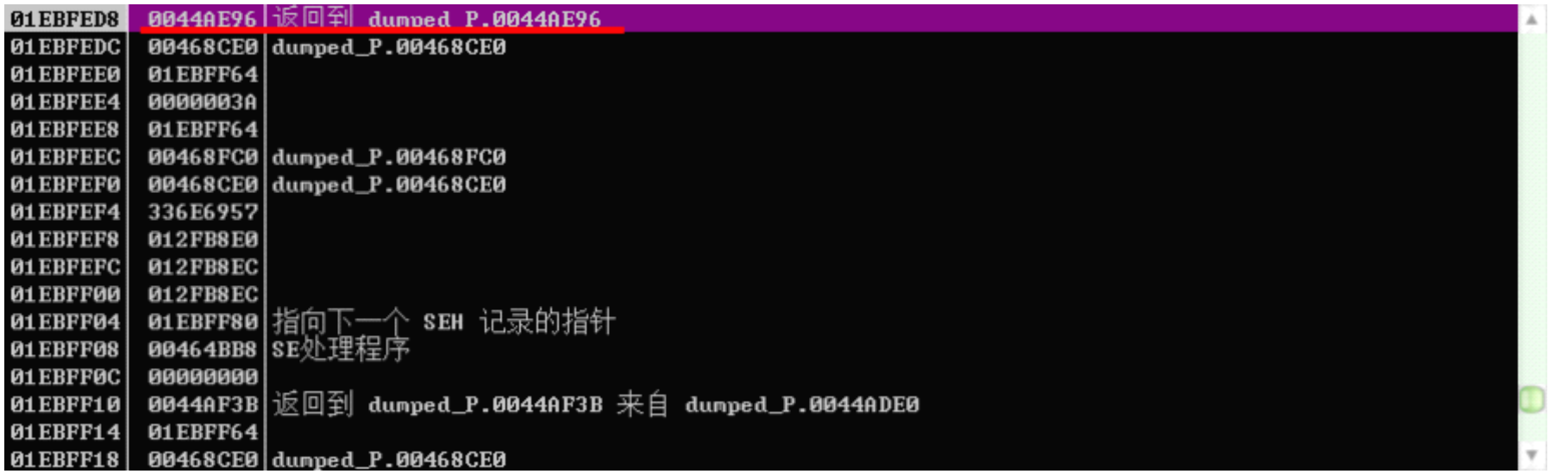
```
00433539  |.  57           push edi
0043353A  |.  B0 72        mov al,0x72
0043353C  |.  884424 2F    mov byte ptr ss:[esp+0x2F],al          ;   填充一个 char 数组
00433540  |.  884424 31    mov byte ptr ss:[esp+0x31],al
00433544  |.  884424 34    mov byte ptr ss:[esp+0x34],al
00433548  |.  884424 39    mov byte ptr ss:[esp+0x39],al
0043354C  |.  884424 3D    mov byte ptr ss:[esp+0x3D],al
00433550  |.  B0 63        mov al,0x63
00433552  |.  884424 40    mov byte ptr ss:[esp+0x40],al
00433556  |.  884424 41    mov byte ptr ss:[esp+0x41],al
0043355A  |.  B0 73        mov al,0x73
0043355C  |.  884424 43    mov byte ptr ss:[esp+0x43],al
00433560  |.  884424 44    mov byte ptr ss:[esp+0x44],al
00433564  |.  B0 6C        mov al,0x6C
00433566  |.  884424 47    mov byte ptr ss:[esp+0x47],al
0043356A  |.  884424 48    mov byte ptr ss:[esp+0x48],al
0043356E  |.  8BB424 A00400>mov esi,dword ptr ss:[esp+0x4A0]
00433575  |.  8B46 0C      mov eax,dword ptr ds:[esi+0xC]         ;   获取目标程序 IMAGE_NT_HEADERS 首地址
00433578  |.  8B56 18      mov edx,dword ptr ds:[esi+0x18]        ;   获取目标程序.text 节
0043357B  |.  B1 6D        mov cl,0x6D
0043357D  |.  884C24 36    mov byte ptr ss:[esp+0x36],cl
00433581  |.  884C24 3E    mov byte ptr ss:[esp+0x3E],cl
00433585  |.  B3 41        mov bl,0x41
00433587  |.  C64424 2C 7B  mov byte ptr ss:[esp+0x2C],0x7B
0043358C  |.  C64424 2D 4F  mov byte ptr ss:[esp+0x2D],0x4F
00433591  |.  C64424 2E 75  mov byte ptr ss:[esp+0x2E],0x75
00433596  |.  C64424 30 50  mov byte ptr ss:[esp+0x30],0x50
0043359B  |.  C64424 32 6F  mov byte ptr ss:[esp+0x32],0x6F
004335A0  |.  C64424 33 67  mov byte ptr ss:[esp+0x33],0x67
004335A5  |.  C64424 35 61  mov byte ptr ss:[esp+0x35],0x61
004335AA  |.  C64424 37 44  mov byte ptr ss:[esp+0x37],0x44
004335AF  |.  C64424 38 69  mov byte ptr ss:[esp+0x38],0x69
004335B4  |.  C64424 3A 7D  mov byte ptr ss:[esp+0x3A],0x7D
004335B9  |.  C64424 3B 5C  mov byte ptr ss:[esp+0x3B],0x5C
004335BE  |.  885C24 3C    mov byte ptr ss:[esp+0x3C],bl
004335C2  |.  885C24 3F    mov byte ptr ss:[esp+0x3F],bl
004335C6  |.  C64424 42 65  mov byte ptr ss:[esp+0x42],0x65
004335CB  |.  C64424 45 2E  mov byte ptr ss:[esp+0x45],0x2E
004335D0  |.  C64424 46 64  mov byte ptr ss:[esp+0x46],0x64
004335D5  |.  C64424 18 4D  mov byte ptr ss:[esp+0x18],0x4D
004335DA  |.  C64424 19 53  mov byte ptr ss:[esp+0x19],0x53
004335DF  |.  C64424 1A 43  mov byte ptr ss:[esp+0x1A],0x43
004335E4  |.  C64424 1B 46  mov byte ptr ss:[esp+0x1B],0x46
004335E9  |.  0FB740 06    movzx eax,word ptr ds:[eax+0x6]        ;   获取区块的总数量
```

```
004335ED  |.  8D0C80        lea ecx,dword ptr ds:[eax+eax*4]          ;  区块总数 * 5 (此处是一个优化，一个区块占40字节，编译器优化为5 * 8)
004335F0  |.  8B6CCA E8     mov ebp,dword ptr ds:[edx+ecx*8-0x18]     ;  ecx * 8为区块所占的空间总数，得到最后一个节在磁盘文件中所占的大小
004335F4  |.  8D44CA D8     lea eax,dword ptr ds:[edx+ecx*8-0x28]     ;  得到最后一个节的首地址
004335F8  |.  8B78 14       mov edi,dword ptr ds:[eax+0x14]           ;  得到最后一个节的磁盘文件偏移
004335FB  |.  8B46 04       mov eax,dword ptr ds:[esi+0x4]            ;  获取到最后一个节末尾偏移
004335FE  |.  03FD          add edi,ebp                               ;  edi 指向了节区的末尾
00433600  |.  8BAC24 9C0400>mov ebp,dword ptr ss:[esp+0x49C]
00433607  |.  8D8F 00390000 lea ecx,dword ptr ds:[edi+0x3900]
0043360D  |.  3BC1          cmp eax,ecx
0043360F  |.  73 1A         jnb Xdumped_P.0043362B
00433611  |.  8B55 20       mov edx,[arg.7]                           ;  取到程序 OEP
00433614  |.  85D2          test edx,edx                              ;  检查 OEP 是否为0
00433616  |.  74 13         je Xdumped_P.0043362B
00433618  |.  8B4E 18       mov ecx,dword ptr ds:[esi+0x18]           ;  获取.text 节的首地址
0043361B  |.  8B79 14       mov edi,dword ptr ds:[ecx+0x14]           ;   .text 节在磁盘中的偏移
0043361E  |.  0379 10       add edi,dword ptr ds:[ecx+0x10]           ;   取.text 节末尾偏移地址
00433621  |.  3BD7          cmp edx,edi                               ;   程序 OEP 与.text 末尾偏移地址比较
00433623  |.  0F82 E4020000 jb dumped_P.0043390D                       ;   OEP 小于.text 末尾偏移地址则跳转，即 OEP 在.text 节中，确定为 vc++6.0

00433629  |.  8BFA          mov edi,edx                               ;   OEP 给到 edi
0043362B  |>  2BC7          sub eax,edi                               ;   eax 为最后一个节的末尾地址，减去 OEP
0043362D  |.  83F8 09       cmp eax,0x9                               ;   OEP 是否在最后一个节区中?
00433630  |.  0F82 D7020000 jb dumped_P.0043390D                       ;   如果 OEP 在最后一个节中,确定为 vc++6.0
00433636  |.  8B16          mov edx,dword ptr ds:[esi]                ;   取到程序在 PEiD 的基址
00433638  |.  8D0C3A        lea ecx,dword ptr ds:[edx+edi]            ;   取到程序在 PEiD 的 OEP 位置
0043363B  |.  8B11          mov edx,dword ptr ds:[ecx]                ;   edx 保留 OEP 处4字节特征
0043363D  |.  81FA 496E7374 cmp edx,0x74736E49                        ;   特征码比较
00433643  |.  75 15         jnz Xdumped_P.0043365A                    ;   不相等则跳转到0043365A 地址处
              ;......
0043365A  |>  81FA 64617461 cmp edx,0x61746164                        ;   特征码比较
00433660  |.  75 27         jnz Xdumped_P.00433689                    ;   不相等则跳转到00433689地址处

              ;    其他识别流程的代码略

00433750  |>  8B46 0C       mov eax,dword ptr ds:[esi+0xC]            ;   获取目标程序 IMAGE_NT_HEADERS 首地址
00433753  |.  8B40 28       mov eax,dword ptr ds:[eax+0x28]           ;   获取目标程序 OEP
00433756  |.  50            push eax                                  ;   压入目标程序 OEP
00433757  |.  8BCE          mov ecx,esi                               ;   this 指针
00433759  |.  E8 727A0100   call dumped_P.0044B1D0                    ;   文件偏移与虚拟地址偏移进行转换
0043375E  |.  8BF8          mov edi,eax                               ;   把修正后的 OEP 赋值给 edi
00433760  |.  8B46 18       mov eax,dword ptr ds:[esi+0x18]           ;   获取.text 节首地址给 eax
00433763  |.  8B48 14       mov ecx,dword ptr ds:[eax+0x14]           ;   获取.text 节的磁盘偏移
00433766  |.  0348 10       add ecx,dword ptr ds:[eax+0x10]           ;   .text 节的磁盘偏移 + 磁盘中所占的大小 = .text 节末尾偏移
00433769  |.  3BF9          cmp edi,ecx                               ;   程序 OEP 与.text 末尾偏移地址比较
```

```
0043376B  |.  0F82 95010000 jb dumped_P.00433906                    ;    OEP 小于.text 末尾偏移地址则跳转，即 OEP 在.text 节中，确定为 vc++6.0

                ;    其他识别流程的代码略

00433906  |>  8BAC24 9C0400>mov ebp,dword ptr ss:[esp+0x49C]
0043390D  |>  6A 18         push 0x18
0043390F  |.  68 E86B4000   push dumped_P.00406BE8                ;  Microsoft Visual C++ 6.0
00433914  |>  8D4D 04       lea ecx,dword ptr ss:[ebp+0x4]
00433917  |.  E8 B4D3FFFF   call dumped_P.00430CD0
0043391C  |.  5F            pop edi
0043391D  |.  5E            pop esi
0043391E  |.  C645 00 01    mov byte ptr ss:[ebp],0x1
00433922  |.  5D            pop ebp
00433923  |.  B0 01         mov al,0x1
00433925  |.  5B            pop ebx
00433926  |.  81C4 88040000 add esp,0x488
0043392C  \.  C3            retn
```

通过上面代码的分析，我们大概知道了判断的流程，先是简单的判断了 OEP 是否在.text 节中，如果 OEP 不在.text 节中，将会判断是否为其他编译器生成的程序，如果条件都不符合，就开始重新获取 OEP 然后通过一个函数对 OEP 进行修正，然后再次比对 OEP 是否存在.text 节，如果符合条件则为 vc++ 6.0 生成的程序。貌似这个判断流程太过于简单，我们就继续跟到上层函数深挖到底吧！

当函数调用后栈顶为返回地址，通过这个我们就知道上层函数了。把断点下在 00433530 处，断下来后我们看看栈顶信息，如下图：



右键跟随到反汇编窗口，代码如下：

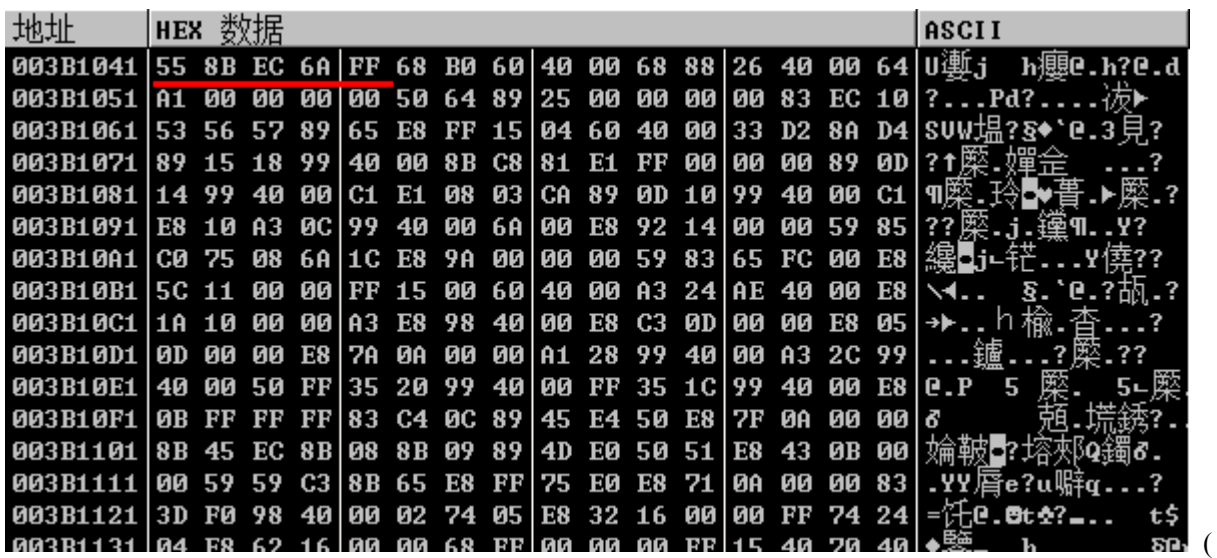
```
0044AE8B  |.  8D0C40        |lea ecx,dword ptr ds:[eax+eax*2]        ;  ecx = eax * 3
0044AE8E  |.  53            |push ebx
0044AE8F  |.  FF148D E43740>|call dword ptr ds:[ecx*4+0x4037E4]      ;  这里是一个函数指针数组寻址调用，ecx 为数组下标
0044AE96  |.  83C4 0C       |add esp,0xC
```

eax 决定了 ecx 的值, ecx 为函数指针数组的下标，所以 eax 的值就是一个关键，断点下在 0044ADE0 处，开始分析该函数，代码如下：

```
0044ADE0  /$  64:A1 0000000>mov eax,dword ptr fs:[0]
0044ADE6  |.  6A FF         push -0x1
0044ADE8  |.  68 B84B4600   push dumped_P.00464BB8                ;  入口地址

0044ADED  |.  50           push eax
0044ADEE  |.  64:8925 00000>mov dword ptr fs:[0],esp              ;  注册异常
0044ADF5  |.  83EC 10       sub esp,0x10
0044ADF8  |.  56           push esi
0044ADF9  |.  8B7424 24     mov esi,dword ptr ss:[esp+0x24]
0044ADFD  |.  8B46 14       mov eax,dword ptr ds:[esi+0x14]       ;  获取 IMAGE_OPTIONAL_HEADER 首地址
0044AE00  |.  8B40 10       mov eax,dword ptr ds:[eax+0x10]       ;  获取目标程序 OEP
0044AE03  |.  57           push edi
0044AE04  |.  8BF9         mov edi,ecx
0044AE06  |.  50           push eax                              ;  压入目标程序 OEP
0044AE07  |.  8BCE         mov ecx,esi                           ;  this 指针
0044AE09  |.  E8 C2030000   call dumped_P.0044B1D0                ;  文件偏移与虚拟地址偏移进行转换
0044AE0E  |.  3B46 04       cmp eax,dword ptr ds:[esi+0x4]        ;  调整后的 OEP 与最后一个节的末尾偏移比较
0044AE11  |.  72 15         jb Xdumped_P.0044AE28                 ;  OEP 小于最后一个节末尾偏移地址则跳转

                ;    失败恢复环境返回上层调用，代码略

0044AE28  |>  53           push ebx
0044AE29  |.  8B5C24 30     mov ebx,dword ptr ss:[esp+0x30]
0044AE2D  |.  8943 20       mov dword ptr ds:[ebx+0x20],eax       ;    保存经过调整的目标程序 OEP
0044AE30  |.  8B4E 04       mov ecx,dword ptr ds:[esi+0x4]        ;    最后一个节的末尾偏移给 ecx
0044AE33  |.  8B16         mov edx,dword ptr ds:[esi]            ;    目标程序基址给到 edx
0044AE35  |.  2BC8         sub ecx,eax
0044AE37  |.  51           push ecx
0044AE38  |.  03D0         add edx,eax
0044AE3A  |.  52           push edx
0044AE3B  |.  8D4C24 14     lea ecx,dword ptr ss:[esp+0x14]
0044AE3F  |.  51           push ecx
0044AE40  |.  8BCF         mov ecx,edi
0044AE42  |.  E8 79740000   call dumped_P.004522C0               ;    程序目标 OEP 数据与特征码经行比较，关键的地方!!
0044AE47  |.  8B4424 14     mov eax,dword ptr ss:[esp+0x14]
0044AE4B  |.  8B4C24 10     mov ecx,dword ptr ss:[esp+0x10]
0044AE4F  |.  8BD0         mov edx,eax
0044AE51  |.  2BD1         sub edx,ecx
0044AE53  |.  C1FA 02       sar edx,0x2
0044AE56  |.  52           push edx
0044AE57  |.  50           push eax
0044AE58  |.  51           push ecx
0044AE59  |.  C74424 30 000>mov dword ptr ss:[esp+0x30],0x0
0044AE61  |.  E8 9AF5FFFF   call dumped_P.0044A400               ;    函数指针数组处理函数的中的下标存进一个数组
```

```
0044AE66  |.  8B7C24 1C     mov edi,dword ptr ss:[esp+0x1C]
0044AE6A  |.  8B4424 20     mov eax,dword ptr ss:[esp+0x20]
0044AE6E  |.  83C4 0C       add esp,0xC
0044AE71  |.  3BF8          cmp edi,eax                              ;    有没有获取到合适的下标
0044AE73  |.  74 37         je Xdumped_P.0044AEAC                    ;    没有获取到合适的下标则跳转
0044AE75  |>  8B07          /mov eax,dword ptr ds:[edi]
0044AE77  |.  50            |push eax
0044AE78  |.  56            |push esi
0044AE79  |.  53            |push ebx
0044AE7A  |.  FF15 E4374000 |call dword ptr ds:[0x4037E4]            ;  dumped_P.004341D0

0044AE80  |.  83C4 0C       |add esp,0xC
0044AE83  |.  84C0          |test al,al
0044AE85  |.  75 48         |jnz Xdumped_P.0044AECF
0044AE87  |.  8B07          |mov eax,dword ptr ds:[edi]
0044AE89  |.  50            |push eax
0044AE8A  |.  56            |push esi
0044AE8B  |.  8D0C40        |lea ecx,dword ptr ds:[eax+eax*2]        ;  取函数指针数组下标
0044AE8E  |.  53            |push ebx
0044AE8F  |.  FF148D E43740>|call dword ptr ds:[ecx*4+0x4037E4]      ;  一个函数指针数组，调用相关函数确认目标程序类型
0044AE96  |.  83C4 0C       |add esp,0xC
0044AE99  |.  84C0          |test al,al
0044AE9B  |.  75 32         |jnz Xdumped_P.0044AECF
0044AE9D  |.  8B4424 14     |mov eax,dword ptr ss:[esp+0x14]
0044AEA1  |.  83C7 04       |add edi,0x4                             ;  调整下标
0044AEA4  |.  3BF8          |cmp edi,eax
0044AEA6  |.^ 75 CD         \jnz Xdumped_P.0044AE75                  ;  函数没有处理，继续循环

                ;    其他代码略
```

有兴趣了解比较过程的可以跟进函数 004522C0，主要是取目标程序 OEP 处数据与特征码数组进行比对，以区分各个编译器或者壳!



地址	HEX 数据												ASCII
003B1041	55	8B	EC	6A	FF	68	B0	60	40	00	68	88	U嫳j h癮@.h?@.d
003B1051	A1	00	00	00	00	50	64	89	25	00	00	00	?...Pd?....泼▶
003B1061	53	56	57	89	65	E8	FF	15	04	60	40	00	SVW塭?§◆`@.3見?
003B1071	89	15	18	99	40	00	8B	C8	81	E1	FF	00	?↑糜.嬋金 ...?
003B1081	14	99	40	00	C1	E1	08	03	CA	89	0D	10	¶糜.玲◘♥賁.▶糜.?
003B1091	E8	10	A3	0C	99	40	00	6A	00	E8	92	14	??糜.j.鑾¶..Y?
003B10A1	C0	75	08	6A	1C	E8	9A	00	00	00	59	83	纔▬j∟钍...Y僥??
003B10B1	5C	11	00	00	FF	15	00	60	40	00	A3	24	\◀.. §.`@.?舐.?
003B10C1	1A	10	00	00	A3	E8	98	40	00	E8	C3	0D	→▶..h榆.查...?
003B10D1	0D	00	00	E8	7A	0A	00	00	A1	28	99	40	...鑪...?糜.??
003B10E1	40	00	50	FF	35	20	99	40	00	FF	35	1C	@.P 5 糜. 5∟糜.
003B10F1	0B	FF	FF	FF	83	C4	0C	89	45	E4	50	E8	♂ 蒐.塭銹?..
003B1101	8B	45	EC	8B	08	8B	09	89	4D	E0	50	51	嫰鞁▬?塔郊q鐲♂.
003B1111	00	59	59	C3	8B	65	E8	FF	75	E0	E8	71	.YY屑e?u嘚q...?
003B1121	3D	F0	98	40	00	02	74	05	E8	32	16	00	=饸@.☻t♣?▂.. t$
003B1131	04	E8	62	16	00	00	68	FF	00	00	00	FF	♦鬣_ h §P▸

(VC++ 6.0 特征码)