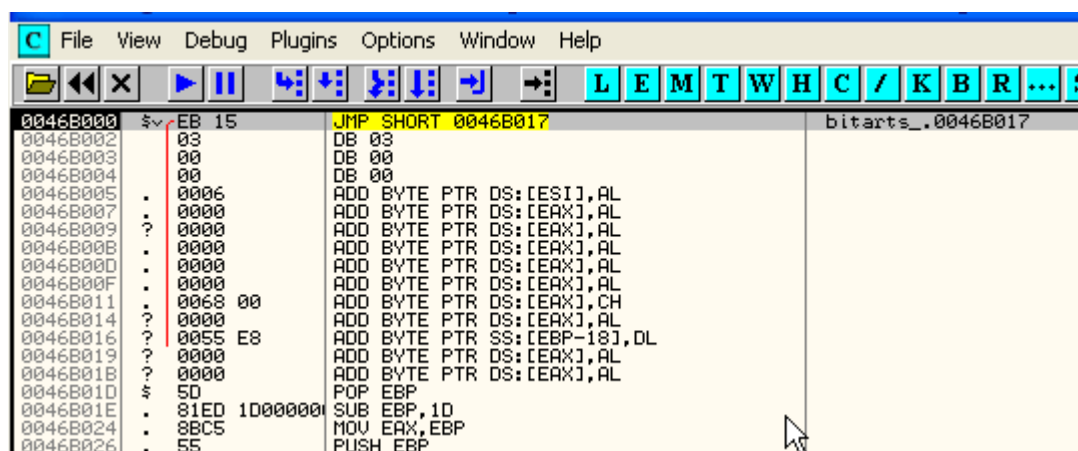


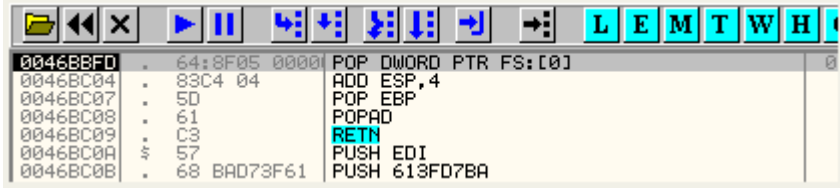
第三十六章-IAT 重定向

本章我们继续增加壳的难度,将看两款壳,分别是 bitarts 5.0, telock 0.98, telock 0.98 会涉及到本章我们要讨论的 IAT 重定向的知识。

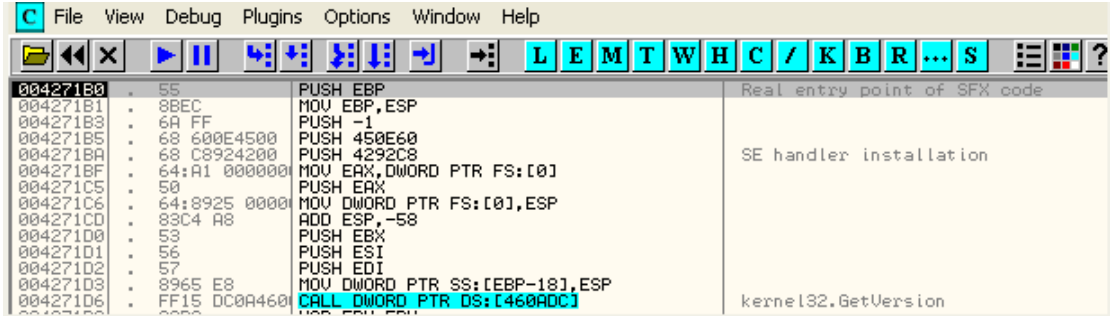
我们先拿简单的 bitarts 5.0 开刀,用 OD 加载它。



按 F9 键运行起来(遇到异常忽略掉),断了下来。



往下单步跟几行就能到达 OEP 处。

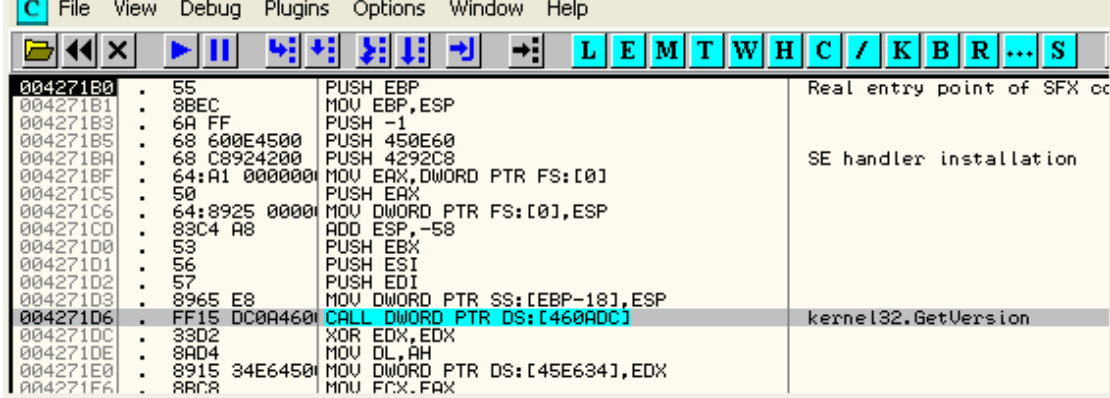


大家也可以使用 OD 自带的功能选项定位 OEP,前面 OEP 章节中介绍的方法大家可以一一尝试。

OEP 命中在第一个区段。

| | | | | | | | | |
|----------|----------|----------|--------|--------------|------|-----|-----|--|
| 003E0000 | 00004000 | | | | Priv | RW | | |
| 003F0000 | 00002000 | | | | Map | R | R | |
| 00400000 | 00001000 | bitarts_ | | PE header | Imag | R | RWE | |
| 00401000 | 0000A000 | bitarts_ | .text | code | Imag | R | RWE | |
| 0044B000 | 0000C000 | bitarts_ | .rdata | | Imag | R | RWE | |
| 00457000 | 00009000 | bitarts_ | .data | data | Imag | R | RWE | |
| 00460000 | 00003000 | bitarts_ | .idata | | Imag | R | RWE | |
| 00463000 | 00008000 | bitarts_ | .rsrc | resources | Imag | R | RWE | |
| 0046B000 | 00008000 | bitarts_ | .edata | SFX, imports | Imag | R | RWE | |
| 004C0000 | 00009000 | | | | Map | R E | R E | |

第一个区段为代码段,OD 不会弹出入口点不在代码段的提示框。



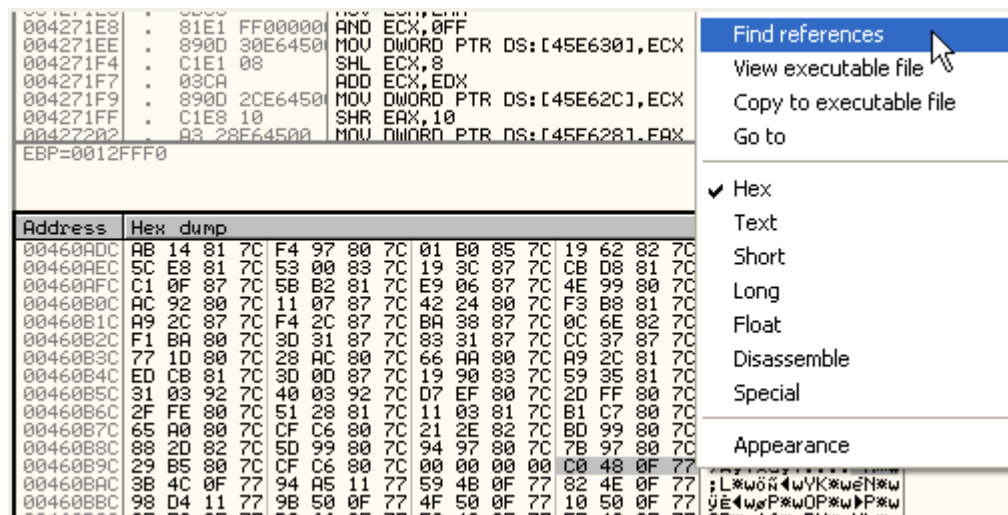
这里我们可以看到调用的第一个 API 函数是 GetVersion,我们可以在一开始给 GetVersion 设置一个断点,运行起来,接着会断下来,看看返回地址是不是位于第一个区段,如果是就定位到返回地址处,上面就是 OEP 了。

定位 OEP 很容易,这里不再赘述了,下面我们来看下 IAT。

460ADC 是 IAT 中的一项,其中保存的是 GetVersion 的地址,我们在数据窗口中定位到 IAT。

| | | | |
|----------|----|--------------|--|
| 00427102 | 57 | PUSH ESI | |
| 00427103 | 58 | MOV ESI, EAX | |
| 00427104 | 59 | MOV ESI, EAX | |
| 00427105 | 5A | MOV ESI, EAX | |
| 00427106 | 5B | MOV ESI, EAX | |
| 00427107 | 5C | MOV ESI, EAX | |
| 00427108 | 5D | MOV ESI, EAX | |
| 00427109 | 5E | MOV ESI, EAX | |
| 0042710A | 5F | MOV ESI, EAX | |
| 0042710B | 60 | MOV ESI, EAX | |
| 0042710C | 61 | MOV ESI, EAX | |
| 0042710D | 62 | MOV ESI, EAX | |
| 0042710E | 63 | MOV ESI, EAX | |
| 0042710F | 64 | MOV ESI, EAX | |
| 00427110 | 65 | MOV ESI, EAX | |
| 00427111 | 66 | MOV ESI, EAX | |
| 00427112 | 67 | MOV ESI, EAX | |
| 00427113 | 68 | MOV ESI, EAX | |
| 00427114 | 69 | MOV ESI, EAX | |
| 00427115 | 6A | MOV ESI, EAX | |
| 00427116 | 6B | MOV ESI, EAX | |
| 00427117 | 6C | MOV ESI, EAX | |
| 00427118 | 6D | MOV ESI, EAX | |
| 00427119 | 6E | MOV ESI, EAX | |
| 0042711A | 6F | MOV ESI, EAX | |
| 0042711B | 70 | MOV ESI, EAX | |
| 0042711C | 71 | MOV ESI, EAX | |
| 0042711D | 72 | MOV ESI, EAX | |
| 0042711E | 73 | MOV ESI, EAX | |
| 0042711F | 74 | MOV ESI, EAX | |
| 00427120 | 75 | MOV ESI, EAX | |
| 00427121 | 76 | MOV ESI, EAX | |
| 00427122 | 77 | MOV ESI, EAX | |
| 00427123 | 78 | MOV ESI, EAX | |
| 00427124 | 79 | MOV ESI, EAX | |
| 00427125 | 7A | MOV ESI, EAX | |
| 00427126 | 7B | MOV ESI, EAX | |
| 00427127 | 7C | MOV ESI, EAX | |
| 00427128 | 7D | MOV ESI, EAX | |
| 00427129 | 7E | MOV ESI, EAX | |
| 0042712A | 7F | MOV ESI, EAX | |
| 0042712B | 80 | MOV ESI, EAX | |
| 0042712C | 81 | MOV ESI, EAX | |
| 0042712D | 82 | MOV ESI, EAX | |
| 0042712E | 83 | MOV ESI, EAX | |
| 0042712F | 84 | MOV ESI, EAX | |
| 00427130 | 85 | MOV ESI, EAX | |
| 00427131 | 86 | MOV ESI, EAX | |
| 00427132 | 87 | MOV ESI, EAX | |
| 00427133 | 88 | MOV ESI, EAX | |
| 00427134 | 89 | MOV ESI, EAX | |
| 00427135 | 8A | MOV ESI, EAX | |
| 00427136 | 8B | MOV ESI, EAX | |
| 00427137 | 8C | MOV ESI, EAX | |
| 00427138 | 8D | MOV ESI, EAX | |
| 00427139 | 8E | MOV ESI, EAX | |
| 0042713A | 8F | MOV ESI, EAX | |
| 0042713B | 90 | MOV ESI, EAX | |
| 0042713C | 91 | MOV ESI, EAX | |
| 0042713D | 92 | MOV ESI, EAX | |
| 0042713E | 93 | MOV ESI, EAX | |
| 0042713F | 94 | MOV ESI, EAX | |
| 00427140 | 95 | MOV ESI, EAX | |
| 00427141 | 96 | MOV ESI, EAX | |
| 00427142 | 97 | MOV ESI, EAX | |
| 00427143 | 98 | MOV ESI, EAX | |
| 00427144 | 99 | MOV ESI, EAX | |
| 00427145 | 9A | MOV ESI, EAX | |
| 00427146 | 9B | MOV ESI, EAX | |
| 00427147 | 9C | MOV ESI, EAX | |
| 00427148 | 9D | MOV ESI, EAX | |
| 00427149 | 9E | MOV ESI, EAX | |
| 0042714A | 9F | MOV ESI, EAX | |
| 0042714B | A0 | MOV ESI, EAX | |
| 0042714C | A1 | MOV ESI, EAX | |
| 0042714D | A2 | MOV ESI, EAX | |
| 0042714E | A3 | MOV ESI, EAX | |
| 0042714F | A4 | MOV ESI, EAX | |
| 00427150 | A5 | MOV ESI, EAX | |
| 00427151 | A6 | MOV ESI, EAX | |
| 00427152 | A7 | MOV ESI, EAX | |
| 00427153 | A8 | MOV ESI, EAX | |
| 00427154 | A9 | MOV ESI, EAX | |
| 00427155 | AA | MOV ESI, EAX | |
| 00427156 | AB | MOV ESI, EAX | |
| 00427157 | AC | MOV ESI, EAX | |
| 00427158 | AD | MOV ESI, EAX | |
| 00427159 | AE | MOV ESI, EAX | |
| 0042715A | AF | MOV ESI, EAX | |
| 0042715B | B0 | MOV ESI, EAX | |
| 0042715C | B1 | MOV ESI, EAX | |
| 0042715D | B2 | MOV ESI, EAX | |
| 0042715E | B3 | MOV ESI, EAX | |
| 0042715F | B4 | MOV ESI, EAX | |
| 00427160 | B5 | MOV ESI, EAX | |
| 00427161 | B6 | MOV ESI, EAX | |
| 00427162 | B7 | MOV ESI, EAX | |
| 00427163 | B8 | MOV ESI, EAX | |
| 00427164 | B9 | MOV ESI, EAX | |
| 00427165 | BA | MOV ESI, EAX | |
| 00427166 | BB | MOV ESI, EAX | |
| 00427167 | BC | MOV ESI, EAX | |
| 00427168 | BD | MOV ESI, EAX | |
| 00427169 | BE | MOV ESI, EAX | |
| 0042716A | BF | MOV ESI, EAX | |
| 0042716B | C0 | MOV ESI, EAX | |
| 0042716C | C1 | MOV ESI, EAX | |
| 0042716D | C2 | MOV ESI, EAX | |
| 0042716E | C3 | MOV ESI, EAX | |
| 0042716F | C4 | MOV ESI, EAX | |
| 00427170 | C5 | MOV ESI, EAX | |
| 00427171 | C6 | MOV ESI, EAX | |
| 00427172 | C7 | MOV ESI, EAX | |
| 00427173 | C8 | MOV ESI, EAX | |
| 00427174 | C9 | MOV ESI, EAX | |
| 00427175 | CA | MOV ESI, EAX | |
| 00427176 | CB | MOV ESI, EAX | |
| 00427177 | CC | MOV ESI, EAX | |
| 00427178 | CD | MOV ESI, EAX | |
| 00427179 | CE | MOV ESI, EAX | |
| 0042717A | CF | MOV ESI, EAX | |
| 0042717B | D0 | MOV ESI, EAX | |
| 0042717C | D1 | MOV ESI, EAX | |
| 0042717D | D2 | MOV ESI, EAX | |
| 0042717E | D3 | MOV ESI, EAX | |
| 0042717F | D4 | MOV ESI, EAX | |
| 00427180 | D5 | MOV ESI, EAX | |
| 00427181 | D6 | MOV ESI, EAX | |
| 00427182 | D7 | MOV ESI, EAX | |
| 00427183 | D8 | MOV ESI, EAX | |
| 00427184 | D9 | MOV ESI, EAX | |
| 00427185 | DA | MOV ESI, EAX | |
| 00427186 | DB | MOV ESI, EAX | |
| 00427187 | DC | MOV ESI, EAX | |
| 00427188 | DD | MOV ESI, EAX | |
| 00427189 | DE | MOV ESI, EAX | |
| 0042718A | DF | MOV ESI, EAX | |
| 0042718B | E0 | MOV ESI, EAX | |
| 0042718C | E1 | MOV ESI, EAX | |
| 0042718D | E2 | MOV ESI, EAX | |
| 0042718E | E3 | MOV ESI, EAX | |
| 0042718F | E4 | MOV ESI, EAX | |
| 00427190 | E5 | MOV ESI, EAX | |
| 00427191 | E6 | MOV ESI, EAX | |
| 00427192 | E7 | MOV ESI, EAX | |
| 00427193 | E8 | MOV ESI, EAX | |
| 00427194 | E9 | MOV ESI, EAX | |
| 00427195 | EA | MOV ESI, EAX | |
| 00427196 | EB | MOV ESI, EAX | |
| 00427197 | EC | MOV ESI, EAX | |
| 00427198 | ED | MOV ESI, EAX | |
| 00427199 | EE | MOV ESI, EAX | |
| 0042719A | EF | MOV ESI, EAX | |
| 0042719B | F0 | MOV ESI, EAX | |
| 0042719C | F1 | MOV ESI, EAX | |
| 0042719D | F2 | MOV ESI, EAX | |
| 0042719E | F3 | MOV ESI, EAX | |
| 0042719F | F4 | MOV ESI, EAX | |
| 004271A0 | F5 | MOV ESI, EAX | |
| 004271A1 | F6 | MOV ESI, EAX | |
| 004271A2 | F7 | MOV ESI, EAX | |
| 004271A3 | F8 | MOV ESI, EAX | |
| 004271A4 | F9 | MOV ESI, EAX | |
| 004271A5 | FA | MOV ESI, EAX | |
| 004271A6 | FB | MOV ESI, EAX | |
| 004271A7 | FC | MOV ESI, EAX | |
| 004271A8 | FD | MOV ESI, EAX | |
| 004271A9 | FE | MOV ESI, EAX | |
| 004271AA | FF | MOV ESI, EAX | |
| 004271AB | | | |
| 004271AC | | | |
| 004271AD | | | |
| 004271AE | | | |
| 004271AF | | | |
| 004271B0 | | | |
| 004271B1 | | | |
| 004271B2 | | | |
| 004271B3 | | | |
| 004271B4 | | | |
| 004271B5 | | | |
| 004271B6 | | | |
| 004271B7 | | | |
| 004271B8 | | | |
| 004271B9 | | | |
| 004271BA | | | |
| 004271BB | | | |
| 004271BC | | | |
| 004271BD | | | |
| 004271BE | | | |
| 004271BF | | | |
| 004271C0 | | | |
| 004271C1 | | | |
| 004271C2 | | | |
| 004271C3 | | | |
| 004271C4 | | | |
| 004271C5 | | | |
| 004271C6 | | | |
| 004271C7 | | | |
| 004271C8 | | | |
| 004271C9 | | | |
| 004271CA | | | |
| 004271CB | | | |
| 004271CC | | | |
| 004271CD | | | |
| 004271CE | | | |
| 004271CF | | | |
| 004271D0 | | | |
| 004271D1 | | | |
| 004271D2 | | | |
| 004271D3 | | | |
| 004271D4 | | | |
| 004271D5 | | | |
| 004271D6 | | | |
| 004271D7 | | | |
| 004271D8 | | | |
| 004271D9 | | | |
| 004271DA | | | |
| 004271DB | | | |
| 004271DC | | | |
| 004271DD | | | |
| 004271DE | | | |
| 004271DF | | | |
| 004271E0 | | | |
| 004271E1 | | | |
| 004271E2 | | | |
| 004271E3 | | | |
| 004271E4 | | | |
| 004271E5 | | | |
| 004271E6 | | | |
| 004271E7 | | | |
| 004271E8 | | | |
| 004271E9 | | | |
| 004271EA | | | |
| 004271EB | | | |
| 004271EC | | | |
| 004271ED | | | |
| 004271EE | | | |
| 004271EF | | | |
| 004271F0 | | | |
| 004271F1 | | | |
| 004271F2 | | | |
| 004271F3 | | | |
| 004271F4 | | | |
| 004271F5 | | | |
| 004271F6 | | | |
| 004271F7 | | | |
| 004271F8 | | | |
| 004271F9 | | | |
| 004271FA | | | |
| 004271FB | | | |
| 004271FC | | | |
| 004271FD | | | |
| 004271FE | | | |
| 004271FF | | | |
| 00427200 | | | |
| 00427201 | | | |
| 00427202 | | | |
| 00427203 | | | |
| 00427204 | | | |
| 00427205 | | | |
| 00427206 | | | |
| 00427207 | | | |
| 00427208 | | | |
| 00427209 | | | |
| 0042720A | | | |
| 0042720B | | | |
| 0042720C | | | |
| 0042720D | | | |
| 0042720E | | | |
| 0042720F | | | |
| 00427210 | | | |
| 00427211 | | | |
| 00427212 | | | |
| 00427213 | | | |
| 00427214 | | | |
| 00427215 | | | |
| 00427216 | | | |
| 00427217 | | | |
| 00427218 | | | |
| 00427219 | | | |
| 0042721A | | | |
| 0042721B | | | |
| 0042721C | | | |
| 0042721D | | | |
| 0042721E | | | |
| 0042721F | | | |
| 00427220 | | | |
| 00427221 | | | |
| 00427222 | | | |
| 00427223 | | | |
| 00427224 | | | |
| 00427225 | | | |
| 00427226 | | | |
| 00427227 | | | |
| 00427228 | | | |
| 00427229 | | | |
| 0042722A | | | |
| 0042722B | | | |
| 0042722C | | | |
| 0042722D | | | |
| 0042722E | | | |
| 0042722F | | | |
| 00427230 | | | |
| 00427231 | | | |
| 00427232 | | | |
| 00427233 | | | |
| 00427234 | | | |
| 00427235 | | | |
| 00427236 | | | |
| 00427237 | | | |
| 00427238 | | | |
| 00427239 | | | |
| 0042723A | | | |
| 0042723B | | | |
| 0042723C | | | |
| 0042723D | | | |
| 0042723E | | | |
| 0042723F | | | |
| 00427240 | | | |
| 00427241 | | | |
| 00427242 | | | |
| 00427243 | | | |
| 00427244 | | | |
| 00427245 | | | |
| 00427246 | | | |
| 00427247 | | | |
| 00427248 | | | |
| 00427249 | | | |
| 0042724A | | | |
| 0042724B | | | |
| 0042724C | | | |
| 0042724D | | | |
| 0042724E | | | |
| 0042724F | | | |
| 00427250 | | | |
| 00427251 | | | |
| 00427252 | | | |
| 00427253 | | | |
| 00427254 | | | |
| 00427255 | | | |
| 00427256 | | | |
| 00427257 | | | |
| 00427258 | | | |
| 00427259 | | | |
| 0042725A | | | |
| 0042725B | | | |
| 0042725C | | | |
| 0042725D | | | |
| 0042725 | | | |

下面我们来看看 77xxxx 这类是属于哪个 DLL 的,我们除了在区段列表表中看,也可以在 770F48C0 这一项上面单击鼠标右键选择
-Find references。



这里可以看到参考引用列表。

| Address | Disassembly | Comment |
|----------|----------------------------|-----------------------|
| 00405435 | CALL DWORD PTR DS:[460BA8] | oleaut32.VariantClear |
| 0041CC2D | CALL DWORD PTR DS:[460BA8] | oleaut32.VariantClear |
| 0041F68C | CALL DWORD PTR DS:[460BA8] | oleaut32.VariantClear |
| 00421197 | CALL DWORD PTR DS:[460BA8] | oleaut32.VariantClear |
| 0042133C | CALL DWORD PTR DS:[460BA8] | oleaut32.VariantClear |
| 004213E5 | CALL DWORD PTR DS:[460BA8] | oleaut32.VariantClear |
| 0042198F | CALL DWORD PTR DS:[460BA8] | oleaut32.VariantClear |
| 00421D90 | CALL DWORD PTR DS:[460BA8] | oleaut32.VariantClear |
| 004477B1 | CALL DWORD PTR DS:[460BA8] | oleaut32.VariantClear |
| 004477D8 | CALL DWORD PTR DS:[460BA8] | oleaut32.VariantClear |
| 004478B1 | CALL DWORD PTR DS:[460BA8] | oleaut32.VariantClear |
| 00447E7D | CALL DWORD PTR DS:[460BA8] | oleaut32.VariantClear |
| 00447F0E | CALL DWORD PTR DS:[460BA8] | oleaut32.VariantClear |
| 00447F67 | CALL DWORD PTR DS:[460BA8] | oleaut32.VariantClear |
| 00495284 | DD bitarts_.00460BA8 | |

这里可以看到基本上都是在第一个区段中调用的 OleAut32.dll 中的 VariantClear 这个 API 函数。

| | | | | | | |
|----------|----------|----------|--------|-------------|-------|-----|
| 76B20000 | 00002000 | winmm | .data | data | Image | RWE |
| 76B22000 | 0000A000 | winmm | .rsrc | resources | Image | RWE |
| 76B2C000 | 00002000 | winmm | .reloc | relocations | Image | RWE |
| 770F0000 | 00001000 | oleaut32 | | PE header | Image | RWE |
| 770F1000 | 0007F000 | oleaut32 | .text | code,import | Image | RWE |
| 77170000 | 00002000 | oleaut32 | .orpc | code | Image | RWE |
| 77172000 | 00003000 | oleaut32 | .data | data | Image | RWE |
| 77175000 | 00001000 | oleaut32 | .rsrc | resources | Image | RWE |
| 77176000 | 00006000 | oleaut32 | .reloc | relocations | Image | RWE |
| 773A0000 | 00001000 | comctl_1 | | PE header | Image | RWE |

区段列表窗口中我们也可以看到 770xxxx 这类地址是属于 OleAut32.dll 的代码段的。

| Address | Hex dump | ASCII |
|----------|---|------------------|
| 00460D9C | 42 8C 01 77 2E 8C 01 77 8B 14 03 77 FE EC 03 77 | B!0w.i0wi0ewu0ew |
| 00460DAC | 83 F7 04 77 DE F2 02 77 DF 1A 03 77 F6 F0 04 77 | 3~Ewi0ewu0ewu0ew |
| 00460DBC | 9C F3 04 77 33 F2 02 77 6C C9 01 77 F6 8B 01 77 | 3~Ew3=ewi0w+i0w |
| 00460DCC | B8 96 01 77 0C 94 01 77 61 C6 03 77 81 E5 02 77 | 000w.00w3ewu0ew |
| 00460DDC | 80 03 08 77 55 E6 01 77 A0 A8 01 77 EA 04 05 77 | 0ewu0w+i0w0ew |
| 00460DEC | 24 13 02 77 58 BF 01 77 33 B9 01 77 65 F6 04 77 | 3!Ew0w0w0ewu0ew |
| 00460DFC | 2F B7 01 77 84 F6 04 77 60 8F 01 77 F5 85 01 77 | 20w+i0w0w0ew |
| 00460E0C | 24 15 03 77 E2 C2 01 77 29 69 05 77 DF BA 05 77 | 3\$ew0w0w1w0ew |
| 00460E1C | 8C 14 02 77 4C 1F 03 77 F9 07 01 77 F7 D6 01 77 | i0ewL0ewu0ewu0ew |
| 00460E2C | 65 C4 01 77 04 B6 01 77 C8 8D 01 77 AE B6 01 77 | e0ew0ewu0ewu0ew |
| 00460E3C | CD 48 02 77 3E 08 02 77 C7 86 01 77 90 86 01 77 | =HEw>0ew0ewu0ew |
| 00460E4C | 26 BF 01 77 3F 85 01 77 69 08 01 77 85 CB 01 77 | 8w0w0ewu0ewu0ew |
| 00460E5C | 71 BE 01 77 6E C6 01 77 90 8F 01 77 F0 BE 01 77 | q0w0w0ew0ewu0ew |
| 00460E6C | 31 B6 01 77 17 E9 03 77 00 23 04 77 9B F3 02 77 | 10w0ewu0ewu0ew |
| 00460E7C | 85 37 02 77 75 8E 01 77 8B EE 04 77 80 00 00 00 | A7ewu0ewu0ewu0ew |
| 00460E8C | F7 A8 81 76 00 00 00 00 C8 74 F8 72 73 66 F9 72 | 0ewu0ewu0ewu0ew |
| 00460E9C | 87 72 F8 72 43 80 F8 72 67 37 F9 72 FB 41 F9 72 | 0ewu0ewu0ewu0ew |
| 00460EAC | 67 83 F8 72 90 53 F8 72 00 00 00 00 CE 00 37 76 | g0ew0ewu0ewu0ew |
| 00460EBC | 7C 86 37 76 80 86 37 76 33 25 36 76 1E 31 36 76 | i0ewu0ewu0ewu0ew |
| 00460ECC | D8 7C 37 76 89 C2 37 76 CD 46 38 76 CE EE 36 76 | i0ewu0ewu0ewu0ew |
| 00460EDC | 00 00 00 00 48 00 4C 77 9C CB 40 77 CC 42 4F 77 | ...H0ewu0ewu0ew |
| 00460EEC | 2C D0 4C 77 0A F6 4C 77 73 33 50 77 10 64 4D 77 | 0ewu0ewu0ewu0ew |
| 00460EFC | 03 0E 52 77 33 0F 52 77 40 A6 54 77 F1 87 54 77 | 0ewu0ewu0ewu0ew |
| 00460F0C | 92 9C 4F 77 6F 57 52 77 93 33 4E 77 B2 5D 4E 77 | 0ewu0ewu0ewu0ew |
| 00460F1C | 90 C0 5A 77 00 00 00 00 F3 F0 CC 74 00 00 00 00 | 0ewu0ewu0ewu0ew |
| 00460F2C | 0B 00 50 60 61 79 53 6F 75 6E 64 41 00 00 57 49 | 0ewu0ewu0ewu0ew |
| 00460F3C | 4E 40 4D 2E 64 6C 6C 00 FE 00 47 65 74 4D 6F 64 | 0ewu0ewu0ewu0ew |
| 00460F4C | 75 6C 65 48 61 6E 64 6C 65 41 00 00 7E 01 49 6E | 0ewu0ewu0ewu0ew |
| 00460F5C | 54 69 72 6C 63 68 65 64 49 6E 63 72 65 6D 60 | 0ewu0ewu0ewu0ew |
| 00460F6C | 6E 74 00 00 75 01 49 6E 74 65 72 6C 63 68 65 | 0ewu0ewu0ewu0ew |
| 00460F7C | 64 44 65 63 72 65 6D 65 6E 74 00 00 9A 01 4C 6F | 0ewu0ewu0ewu0ew |
| 00460F8C | 63 61 6C 46 72 65 65 00 9C 01 4C 6F 63 61 6C 4C | 0ewu0ewu0ewu0ew |
| 00460F9C | 6F 63 68 00 96 01 4C 6F 63 61 6C 41 6C 6C 6F 63 | 0ewu0ewu0ewu0ew |
| 00460FAC | 00 00 80 01 4C 6F 63 61 6C 55 6E 6C 6F 63 68 00 | 0ewu0ewu0ewu0ew |

这里我们就不一一查看其他 DLL 的 IAT 项了,直接定位到 IAT 的尾部。

| | | | | | |
|----------|-------------|-------------|-------------|-------------|------------------|
| 00460EBC | 7C 86 37 76 | B0 86 37 76 | 33 25 36 76 | 1E 31 36 76 | 137v37v3%6v16v |
| 00460ECC | 08 7C 37 76 | 89 C2 37 76 | CD 46 38 76 | CE EE 36 76 | i17v87vF8v47v6v |
| 00460EDC | 00 00 00 00 | 48 D0 4C 77 | 9C CB 4D 77 | CC 42 4F 77 | ...H3Lw3Pw1B0w |
| 00460EEC | 2C D0 4C 77 | DA F6 4C 77 | 73 33 50 77 | 10 64 4D 77 | ,\$Lw3Pw1d1w |
| 00460EFC | 03 0E 52 77 | 33 0F 52 77 | 40 A6 54 77 | F1 A7 54 77 | 00Rw3Rw3Tw10Tw |
| 00460F0C | 92 9C 4F 77 | 6F 57 52 77 | 99 33 4E 77 | B2 5D 4E 77 | E00w0Rw03Nw3JNw |
| 00460F1C | 90 C0 5A 77 | 00 00 00 00 | F3 F0 CC 74 | 00 00 00 00 | E'2w...%-1ft... |
| 00460F2C | 0B 00 50 6C | 61 79 53 6F | 75 61 64 41 | 00 00 57 49 | 0.PlaySoundA..MI |
| 00460F3C | 4E 4D 4D 2E | 64 6C 6C 00 | FE 00 47 65 | 74 4D 6F 64 | NMM.dll. GetMod |
| 00460F4C | 75 6C 65 48 | 61 6E 64 6C | 65 41 00 00 | 7E 01 49 6E | uleHandleA..0In |
| 00460F5C | 74 65 72 6C | 6F 63 68 65 | 64 41 6E 63 | 72 65 6D 65 | terlockedIncreme |
| 00460F6C | 6E 74 00 00 | 7B 01 49 6E | 74 61 72 6C | 6F 63 6B 65 | nt..0Interlocke |
| 00460F7C | 64 44 65 63 | 72 65 6D 65 | 6E 74 00 00 | 9A 01 4C 6F | dDecrement..00Lo |
| 00460F8C | 63 61 6C 46 | 72 65 65 00 | 9C 01 4C 6F | 63 61 6C 4C | calFree.00LocalL |
| 00460F9C | 6F 63 6B 00 | 96 01 4C 6F | 63 61 6C 41 | 6C 6C 6F 63 | ock.00LocalAtLoc |
| 00460FAC | 00 00 A0 01 | 4C 6F 63 61 | 6C 55 6E 6C | 6F 63 6B 00 | ..00LocalUnlock. |
| 00460FBC | A3 01 4C 6F | 63 6B 52 65 | 73 6F 75 72 | 63 65 00 00 | 00LockResource.. |

这里用湛蓝色, 粉红色,灰色分别标注了不同 DLL 的 IAT 项,分割的零用绿色标注出来,在最后一个 IAT 项上面单击鼠标右键选择 -Find references。

| Address | Hex dump |
|----------|-------------------------------------|
| 00460D9C | 42 8C D1 77 2E 8C D1 77 8B 14 D3 77 |
| 00460DAC | 83 F7 04 77 DE F2 02 77 DF 1A D3 77 |
| 00460DBC | 9C F3 04 77 33 F2 02 77 6C C9 D1 77 |
| 00460DCC | B8 96 01 77 0C 94 D1 77 61 C6 D3 77 |
| 00460DDC | 80 03 D3 77 55 E6 D1 77 AD A8 D1 77 |
| 00460DEC | 24 13 D2 77 58 BF D1 77 33 B9 D1 77 |
| 00460DFC | 2F B7 D1 77 B4 F6 04 77 6C BF D1 77 |
| 00460E0C | 24 15 D3 77 E2 C2 D1 77 29 69 D5 77 |
| 00460E1C | 8C 14 D2 77 4C 1F D3 77 F9 D7 D1 77 |
| 00460E2C | 65 C4 D1 77 D4 B6 D1 77 C8 BD D1 77 |
| 00460E3C | CD 48 D2 77 3E 0B D2 77 C7 86 D1 77 |
| 00460E4C | 26 BF D1 77 3F B5 D1 77 69 D8 D1 77 |
| 00460E5C | 71 BE D1 77 6E C6 D1 77 9D 8F D1 77 |
| 00460E6C | 31 B6 D1 77 17 E9 D3 77 00 EE D4 77 |
| 00460E7C | B5 37 D2 77 78 8E D1 77 8B EE D4 77 |
| 00460E8C | F7 A8 B1 76 00 00 00 00 C8 74 F8 72 |
| 00460E9C | 87 72 F8 72 43 80 F8 72 67 37 F9 72 |
| 00460EAC | 67 83 F8 72 90 53 F8 72 00 00 00 00 |
| 00460EBC | 7C 86 37 76 B0 86 37 76 33 25 36 76 |
| 00460ECC | D8 7C 37 76 89 C2 37 76 CD 46 38 76 |
| 00460EDC | 00 00 00 00 48 D0 4C 77 9C CB 4D 77 |
| 00460EEC | 2C D0 4C 77 DA F6 4C 77 73 33 50 77 |
| 00460EFC | 03 0E 52 77 33 0F 52 77 40 A6 54 77 |
| 00460F0C | 92 9C 4F 77 6F 57 52 77 99 33 4E 77 |
| 00460F1C | 90 C0 5A 77 00 00 00 00 F3 F0 CC 74 |
| 00460F2C | 0B 00 50 6C 61 79 53 6F 75 6E 64 41 |
| 00460F3C | 4E 4D 4D 2E 64 6C 6C 00 FE 00 47 65 |

Follow DWORD in Dump
 Find references
 View executable file
 Copy to executable file
 Go to
 Hex
 Text
 Short
 Long
 Float
 Disassemble
 Special
 Appearance

| Address | Disassembly | Comment |
|----------|---------------------------|-------------------|
| 00435D38 | JMP DWORD PTR DS:[460F24] | oledlg.OleUIBusyA |

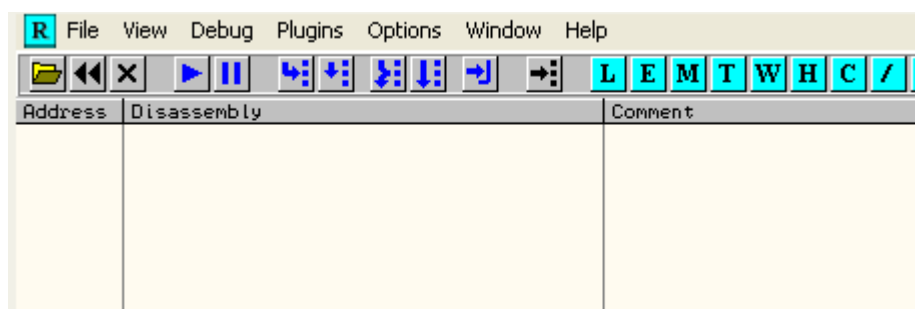
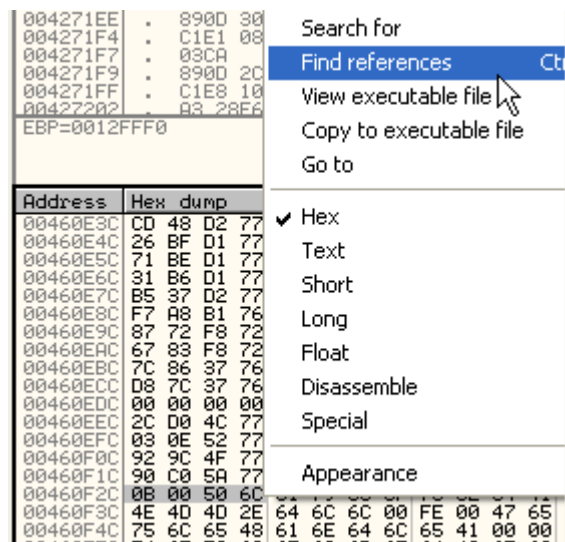
我们可以看到最后一个 IAT 项是属于 oledlg.dll 中 OleUiBushA 这个 API 函数,我们在区段列表窗口中来验证一下。

| | | | | | | | |
|----------|----------|----------|--------|-------------|-------|---|-----|
| 72FA4000 | 00002000 | winspool | .reloc | relocations | Image | R | RWE |
| 74CC0000 | 00010000 | oledlg | | PE header | Image | R | RWE |
| 74CC1000 | 00011000 | oledlg | .text | code,import | Image | R | RWE |
| 74CD2000 | 00002000 | oledlg | .data | data | Image | R | RWE |
| 74CD3000 | 0000B000 | oledlg | .rsrc | resources | Image | R | RWE |
| 74CD4000 | 00001000 | oledlg | .reloc | relocations | Image | R | RWE |
| 76360000 | 00001000 | comdlg32 | | PE header | Image | R | RWE |

下面的几项都不是 IAT 项了。

| | | | | | |
|----------|-------------|-------------|-------------|-------------|------------------|
| 00460EBC | 7C 86 37 76 | B0 86 37 76 | 33 25 36 76 | 1E 31 36 76 | 137v37v3%6v16v |
| 00460ECC | 08 7C 37 76 | 89 C2 37 76 | CD 46 38 76 | CE EE 36 76 | i17v87vF8v47v6v |
| 00460EDC | 00 00 00 00 | 48 D0 4C 77 | 9C CB 4D 77 | CC 42 4F 77 | ...H3Lw3Pw1B0w |
| 00460EEC | 2C D0 4C 77 | DA F6 4C 77 | 73 33 50 77 | 10 64 4D 77 | ,\$Lw3Pw1d1w |
| 00460EFC | 03 0E 52 77 | 33 0F 52 77 | 40 A6 54 77 | F1 A7 54 77 | 00Rw3Rw3Tw10Tw |
| 00460F0C | 92 9C 4F 77 | 6F 57 52 77 | 99 33 4E 77 | B2 5D 4E 77 | E00w0Rw03Nw3JNw |
| 00460F1C | 90 C0 5A 77 | 00 00 00 00 | F3 F0 CC 74 | 00 00 00 00 | E'2w...%-1ft... |
| 00460F2C | 0B 00 50 6C | 61 79 53 6F | 75 6E 64 41 | 00 00 57 49 | 0.PlaySoundA..MI |
| 00460F3C | 4E 4D 4D 2E | 64 6C 6C 00 | FE 00 47 65 | 74 4D 6F 64 | NMM.dll. GetMod |
| 00460F4C | 75 6C 65 48 | 61 6E 64 6C | 65 41 00 00 | 7E 01 49 6E | uleHandleA..0In |
| 00460F5C | 74 65 72 6C | 6F 63 68 65 | 64 41 6E 63 | 72 65 6D 65 | terlockedIncreme |
| 00460F6C | 6E 74 00 00 | 7B 01 49 6E | 74 61 72 6C | 6F 63 6B 65 | nt..0Interlocke |
| 00460F7C | 64 44 65 63 | 72 65 6D 65 | 6E 74 00 00 | 9A 01 4C 6F | dDecrement..00Lo |
| 00460F8C | 63 61 6C 46 | 72 65 65 00 | 9C 01 4C 6F | 63 61 6C 4C | calFree.00LocalL |
| 00460F9C | 6F 63 6B 00 | 96 01 4C 6F | 63 61 6C 41 | 6C 6C 6F 63 | ock.00LocalAtLoc |
| 00460FAC | 00 00 A0 01 | 4C 6F 63 61 | 6C 55 6E 6C | 6F 63 6B 00 | ..00LocalUnlock. |
| 00460FBC | A3 01 4C 6F | 63 6B 52 65 | 73 6F 75 72 | 63 65 00 00 | 00LockResource.. |

我们在 6C5000B 上面单击鼠标右键选择-Find references。



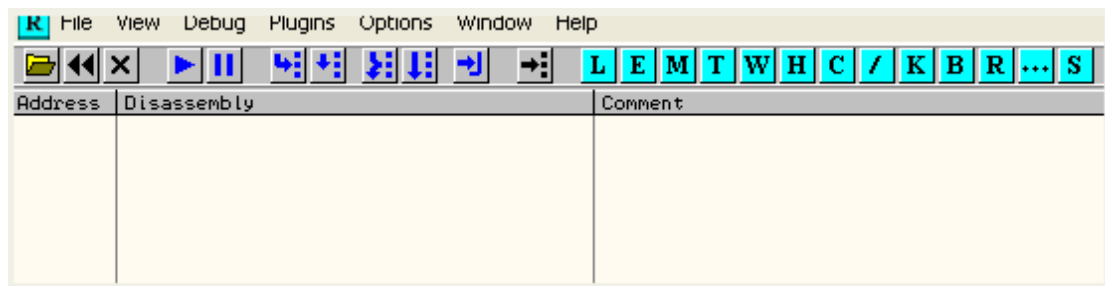
可以看到参考引用列表是空的,所以说 IAT 的最后一项的起始地址为 460F28。

| Address | Hex dump | ASCII |
|----------|---|-------------------|
| 00460EF8 | 10 64 40 77 03 0E 52 77 33 0F 52 77 40 A6 54 77 | !dMw*RW3*RW@TW |
| 00460F08 | F1 A7 54 77 92 9C 4F 77 6F 57 52 77 99 33 4E 77 | !TWfE&QwoloRWd3Hw |
| 00460F18 | B2 5D 4E 77 90 C0 5A 77 00 00 00 00 F3 F0 CC 74 | !HwE'2w...%-ft |
| 00460F28 | 00 00 00 00 00 00 50 6C 61 79 53 6F 75 6E 64 41 |0.PlaySoundA |
| 00460F38 | 00 00 57 49 4E 4D 4D 2E 64 6C 6C 00 FE 00 47 65 | ..WINMM.dll..Ge |
| 00460F48 | 74 4D 6F 64 75 6C 65 48 61 6E 64 6C 65 41 00 00 | tModuleHandleA.. |
| 00460F58 | 7E 01 49 6E 74 65 72 6C 6F 63 6B 65 64 49 6E 63 | ~0InterlockedInc |

我们再来看看哪里是 IAT 的起始位置。

| Address | Hex dump | ASCII |
|----------|---|---------------------|
| 004607C0 | 80 28 06 00 70 28 06 00 00 00 00 00 0C 2B 06 00 | C(+.p(+.....+. |
| 004607D0 | FA 2A 06 00 E8 2A 06 00 1E 2B 06 00 08 2A 06 00 | .*.p*.A+*.i*. |
| 004607E0 | C6 3A 06 00 AE 2A 06 00 92 2A 06 00 72 2A 06 00 | g*..<*..E*..r*.. |
| 004607F0 | BC 3A 06 00 A8 2B 06 00 92 2B 06 00 78 2B 06 00 | +*.c+*.E+*.x+*. |
| 00460800 | 60 3A 06 00 4C 2B 06 00 2E 2B 06 00 00 00 00 00 | *+*.L+*..+*..... |
| 00460810 | 08 00 00 00 00 00 00 00 F0 6B DA 77 1B 76 DA 77 | 0..0....-k rw+Vrw |
| 00460820 | F4 EA DA 77 E7 EB DA 77 83 78 DA 77 00 00 00 00 | !U rWpU rWax rw.... |
| 00460830 | DD 15 C5 58 2E BD C3 58 00 00 00 00 04 6A EF 77 | !S+X.c+X...Ej'w |
| 00460840 | 66 95 EF 77 89 6A EF 77 F3 AD EF 77 ED 09 EF 77 | f0'wEj'w&i'wY'w |
| 00460850 | 99 8B EF 77 C0 B5 EF 77 2A 7D EF 77 B2 7C EF 77 | 0i'wL'A'w*}wW!w |
| 00460860 | 77 53 F2 77 1E C9 F1 77 0C BC EF 77 52 D4 EF 77 | WS=wAfz'w.'wRE'w |
| 00460870 | FA 8D EF 77 F1 DD EF 77 51 B2 EF 77 26 D5 EF 77 | .i'wz!'wWw'w&'w |
| 00460880 | 2A E3 EF 77 8F 39 F2 77 71 B4 EF 77 2E AD EF 77 | *0'w_9=wqf'w.i'w |
| 00460890 | E1 61 EF 77 B8 85 EF 77 CC D2 EF 77 43 70 EF 77 | pa'w@a'wlfE'wCp'w |
| 004608A0 | F8 EA F0 77 12 83 EF 77 01 72 F0 77 A9 34 F0 77 | 'U-w@a'w0r-w04-w |
| 004608B0 | D5 93 EF 77 68 EF EF 77 AA D2 EF 77 B2 6F EF 77 | '0'wh''w-e'wW0'w |
| 004608C0 | 3F 38 F2 77 D6 E8 EF 77 68 E0 EF 77 00 60 EF 77 | ?8=wip'wh0'w.'w |
| 004608D0 | 90 5B EF 77 6D AC EF 77 94 6C F0 77 22 8D EF 77 | E['wm%w0-l-w''l'w |
| 004608E0 | 3D C8 F1 77 3D 6D F0 77 6F C0 EF 77 85 7B EF 77 | =Bzwm-m-woL'w@c'w |
| 004608F0 | 26 D9 EF 77 FB 5E EF 77 36 8A EF 77 FC 8A EF 77 | &'wI^'w6e'w'2'e'w |
| 00460900 | 0F 62 EF 77 49 5E EF 77 97 5D EF 77 1A 9A EF 77 | *b'wI^'wu]w+u'w |

这里可以看到 460810 里面的值是 80000008,明显不会属于任何一个 DLL,我们可以在查看一下其参考引用列表来验证一下,在其上面单击鼠标右键选择-Find references。

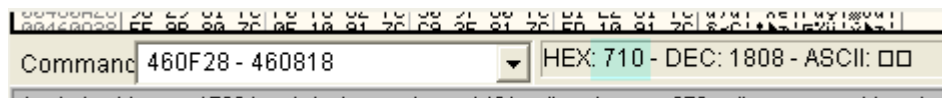


可以看到参考引用列表也是空的,说明 IAT 的第一项的起始地址为 406818。

| Address | Hex dump | ASCII |
|----------|---|------------------------|
| 004607D8 | 1E 2B 06 00 08 2A 06 00 C6 2A 06 00 AE 2A 06 00 | ▲+*.i**.*g**.*<***. |
| 004607E8 | 92 2A 06 00 72 2A 06 00 BC 2B 06 00 A8 2B 06 00 | IE**.*r**.*" +*.c +*. |
| 004607F8 | 92 2B 06 00 78 2B 06 00 60 2B 06 00 4C 2B 06 00 | IE+*.x+*.*'+*.*L+*. |
| 00460808 | 2E 2B 06 00 00 00 00 00 08 00 00 00 00 00 00 00 | .+*.*.....[.C..... |
| 00460818 | F0 68 DA 77 1B 76 DA 77 F4 EA DA 77 E7 EB DA 77 | =k rw+U rw9U rWpU rW |
| 00460828 | 83 78 DA 77 00 00 00 00 DD 15 C5 58 2E BD C3 58 | âx rw....!S+X.c tX |
| 00460838 | 00 00 00 00 04 6A EF 77 66 95 EF 77 89 6A EF 77 | ...Ej'wfo'wEj'w |
| 00460848 | F3 AD EF 77 ED D9 EF 77 99 8B EF 77 C0 B5 EF 77 | %i'wY'w0 i'w lA'w |
| 00460858 | 2A 7D EF 77 B2 7C EF 77 77 53 F2 77 1E C9 F1 77 | *)'wW! 'wWS=wA f:w |
| 00460868 | 0C BC EF 77 52 D4 EF 77 FA 8D EF 77 F1 DD EF 77 | .d'wRE'w. l'w t! 'w |
| 00460878 | 51 B2 EF 77 26 D5 EF 77 2A E3 EF 77 5F 39 F2 77 | CW'w&'w*D'w_9=w |
| 00460888 | 71 B4 EF 77 2E AD EF 77 E1 61 EF 77 B8 85 EF 77 | q l'w. i'w p a'w @ a'w |
| 00460898 | CC D2 EF 77 43 70 EF 77 FB EA F0 77 12 83 EF 77 | l fE'wCp'w' U -w# a'w |
| 004608A8 | 01 72 F0 77 A9 34 F0 77 D5 93 EF 77 68 EF EF 77 | 0x-w@4-w'ô'wh'w |
| 004608B8 | AA D2 EF 77 B2 6F EF 77 3F 38 F2 77 D6 E8 EF 77 | -E'wW o'w?8=wip'w |
| 004608C8 | 68 E0 EF 77 00 60 EF 77 90 5B EF 77 6D AC EF 77 | h0'w.'wE['wm%w |
| 004608D8 | 94 6C F0 77 22 8D EF 77 3D C8 F1 77 3D 6D F0 77 | ô l-w''i'w= t:w=m-w |
| 004608E8 | 6F C0 EF 77 85 7B EF 77 26 D9 EF 77 FB 5E EF 77 | o l'wâ['w&'w l^w |
| 004608F8 | 36 8A EF 77 FC 8A EF 77 0F 62 EF 77 49 5E EF 77 | 6e'w? e'w# b'w l^w |
| 00460908 | 97 5D EF 77 1A 9A EF 77 6B FA EF 77 7B C9 F0 77 | û j'w+U'wk.'w[f-w |
| 00460918 | DA 98 F2 77 1A 40 F2 77 55 EA EF 77 C5 61 EF 77 | ry=w+@=wU0'w t a'w |
| 00460928 | 70 E6 EF 77 F0 81 EF 77 2D 6C EF 77 98 6E EF 77 | pu'w-u'w-l'wyn'w |

现在我们知道了 IAT 的起始地址和结束位置,我们来计算一下 IAT 的大小:

IAT 大小 = 460F28 - 460818 = 710。



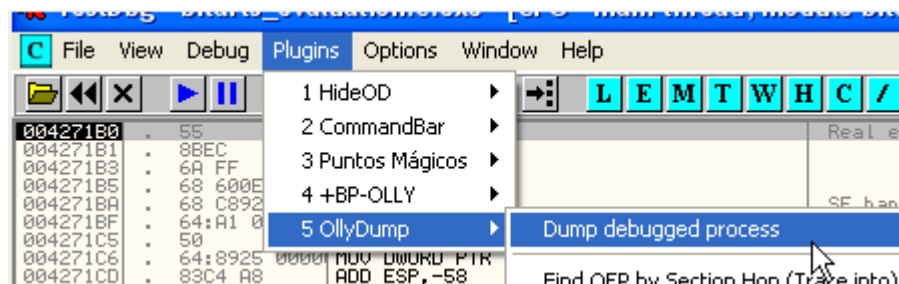
IMP REC 需要的三条数据:

OEP = 271B0(RVA)

IAT 起始地址 = 60818(RVA)

IAT 大小 = 710

下来我们用 OllyDump 插件来 dump。



OllyDump - bitarts_evaluation.c.exe

Start Address: 400000 Size: B3000 **Dump**

Entry Point: 68000 -> Modify: 271B0 Get EIP as OEP Cancel

Base of Code: 1000 Base of Data: 4B000

☒ Fix Raw Size & Offset of Dump Image

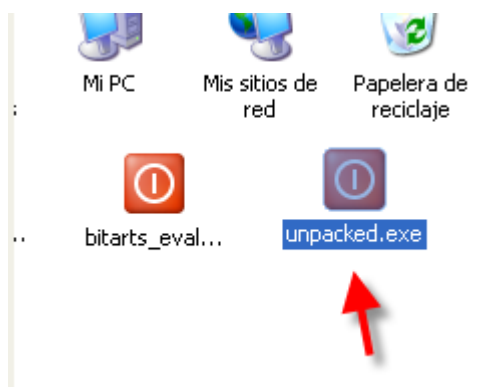
| Section | Virtual Size | Virtual Offset | Raw Size | Raw Offset | Characteristics |
|---------|--------------|----------------|----------|------------|-----------------|
| .text | 0004A000 | 00001000 | 0004A000 | 00001000 | E0000020 |
| .rdata | 0000C000 | 0004B000 | 0000C000 | 0004B000 | C0000040 |
| .data | 00008BE4 | 00057000 | 00008BE4 | 00057000 | C0000040 |
| .idata | 00003000 | 00060000 | 00003000 | 00060000 | C0000040 |
| .rsrc | 00008000 | 00063000 | 00008000 | 00063000 | C0000040 |
| .edata | 000477EE | 0006B000 | 000477EE | 0006B000 | E0000020 |
| | | | | | |
| | | | | | |

☐ Rebuild Import

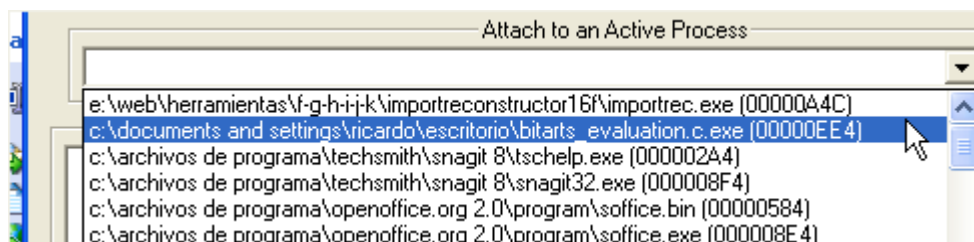
☒ Method1 : Search JMP[API] | CALL[API] in memory image

☐ Method2 : Search DLL & API name string in dumped file

这里不够选 Rebuild Import, 仅仅 dump。



打开 IMP REC, 定位到 bitarts 所在的进程, 当前该进程处于 OEP 处。



将 OEP, RVA, Size 的值都设置上。

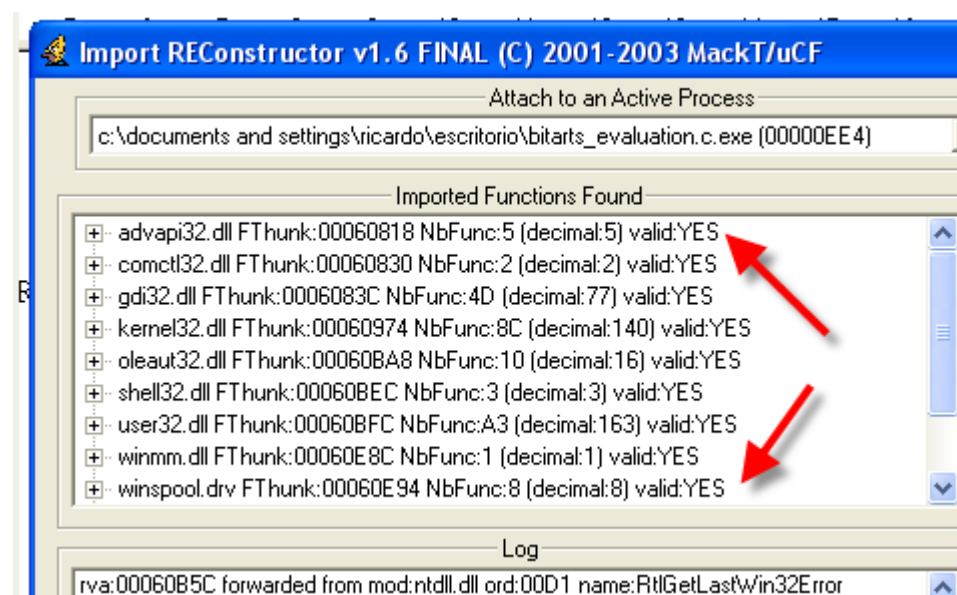
IAT Infos needed

OEP 000271B0 IAT AutoSearch

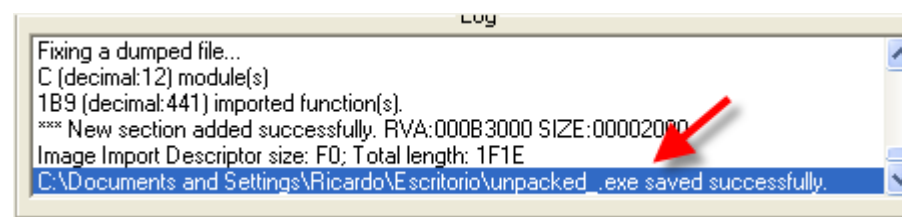
RVA 00060818 Size 00000710

Load Tree Save Tree Get Imports

单击 Get Imports。



可以看到该壳在 IAT 中添加垃圾数据,修复的 IAT 项都是有效的,单击 Fix Dump,选择刚刚 dump 出来的文件修复之。

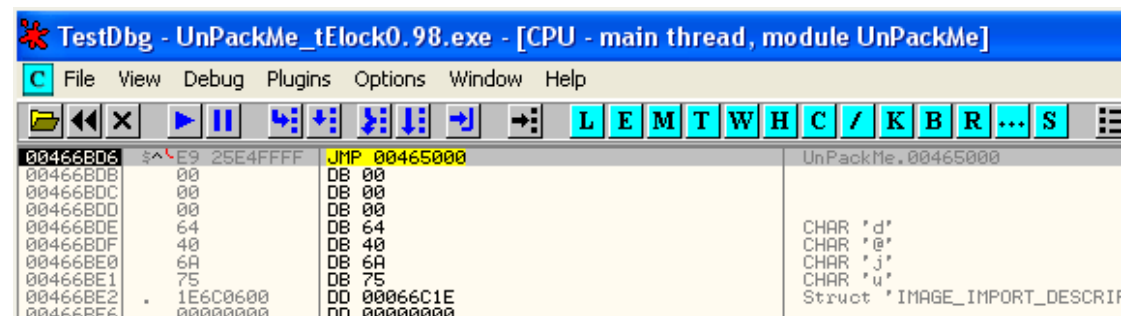


修复过的程序被命名为 unpacked_.exe,运行一下看看效果。



完美运行,没有 AntiDump,比之前介绍的壳稍微复杂一点点。

下面我们来看看 telock 0.98 这款壳,该壳涉及到了我们将要介绍的 IAT 重定向的知识点。



这里我们依然用 ESP 定律来定位 OEP,单步跟几步就能看到 PUSHAD 指令。

| | | | |
|----------|---------------|-------------------|-------------------|
| 00465000 | > 90 | NOP | |
| 00465001 | . 60 | PUSHAD | |
| 00465002 | . E8 02000000 | CALL 00465009 | UnPackMe.00465009 |
| 00465007 | E8 00 | DB E8 | |
| 00465008 | 00 | DB 00 | |
| 00465009 | . E8 00000000 | CALL 0046500E | UnPackMe.0046500E |
| 0046500E | . 5E | POP ESI | |
| 0046500F | . 2BC9 | SUB ECX,ECX | |
| 00465011 | . 58 | POP EAX | |
| 00465012 | . 74 02 | JE SHORT 00465016 | UnPackMe.00465016 |
| 00465014 | . CD 20 | INT 20 | |

按 F7 键执行 PUSHAD 指令,接着在寄存器窗口中定位到 ESP 寄存器的值,在其上面单击鼠标右键选择-Follow in Dump。

| Address | Hex dump | ASCII |
|----------|-------------|--------|
| 0012FFB4 | 38 07 92 70 | Backup |
| 0012FFB4 | 00 E0 FD 7F | Copy |
| 0012FFC4 | 4F 6D 81 70 | Binary |
| 0012FFD4 | 38 A9 54 80 | |
| 0012FFE4 | F3 99 83 70 | |
| 0012FFF4 | 00 00 00 00 | |

| | |
|------------------------------|------------------------|
| Breakpoint | Memory, on access |
| Search for | Memory, on write |
| Follow DWORD in Disassembler | Hardware, on access |
| Follow DWORD in Dump | Hardware, on write |
| Go to | Hardware, on execution |

| |
|-------|
| Byte |
| Word |
| Dword |

给前 4 个字节设置硬件访问断点,运行起来。

| | | | |
|----------|----|-------|----------|
| 004650A3 | 90 | DB 90 | |
| 004650A4 | 90 | DB 90 | |
| 004650A5 | 33 | DB 33 | CHAR '3' |
| 004650A6 | DB | DB DB | |
| 004650A7 | F7 | DB F7 | |
| 004650A8 | F3 | DB F3 | |
| 004650A9 | 64 | DB 64 | CHAR 'd' |
| 004650AA | 67 | DB 67 | CHAR 'g' |
| 004650AB | 8F | DB 8F | |
| 004650AC | 06 | DB 06 | |
| 004650AD | 00 | DB 00 | |
| 004650AE | 00 | DB 00 | |

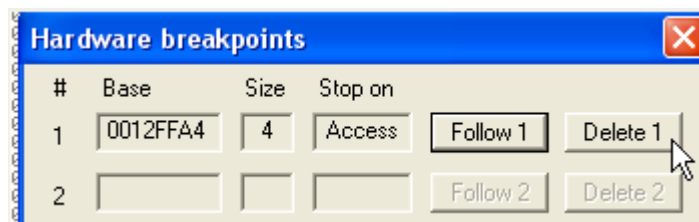
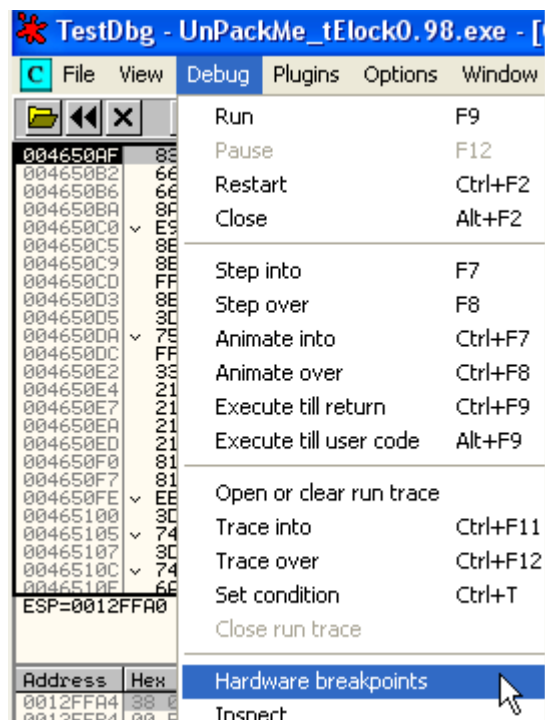
单击鼠标右键选择-Analysis-Remove analysis from module 删除掉 OD 的分析结果。

| | |
|-----------------------|-----------------------------|
| Copy to executable | Analyse code |
| Analysis | Remove analysis from module |
| Dump debugged process | Scan object files |

我们可以看到断在了这里。

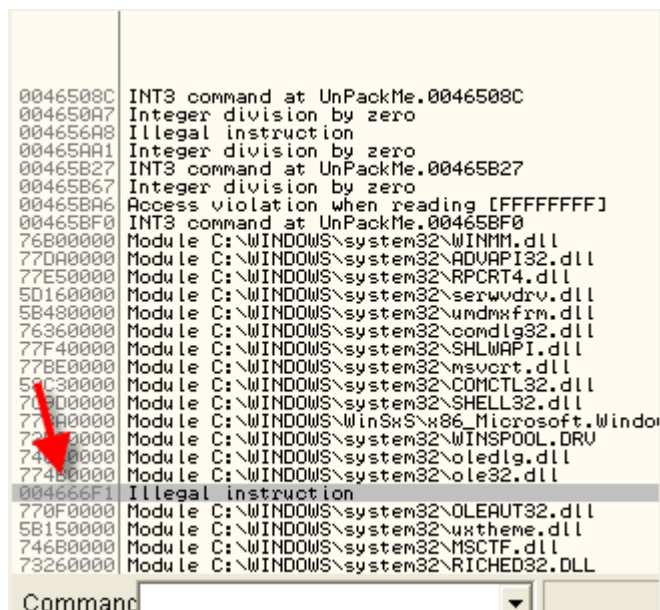
| | | | |
|----------|-----------------|-------------------------------|-------------------|
| 004650A0 | C1C0 07 | ROL EAX,7 | |
| 004650A3 | 90 | NOP | |
| 004650A4 | 90 | NOP | |
| 004650A5 | 33DB | XOR EBX,EBX | |
| 004650A7 | F7F3 | DIU EBX | |
| 004650A9 | 64:67:8F06 0000 | POP DWORD PTR FS:[0] | |
| 004650AF | 83C4 04 | ADD ESP,4 | |
| 004650B2 | 66:BE 4746 | MOV SI,4647 | |
| 004650B6 | 66:BF 4D4A | MOV DI,4A4D | |
| 004650BA | 8A85 99000000 | MOV AL,BYTE PTR SS:[EBP+99] | |
| 004650C0 | E9 9C000000 | JMP 00465161 | UnPackMe.00465161 |
| 004650C5 | 8B4424 04 | MOV EAX,DWORD PTR SS:[ESP+4] | |
| 004650C9 | 8B4C24 0C | MOV ECX,DWORD PTR SS:[ESP+C] | |
| 004650CD | FF81 B8000000 | INC DWORD PTR DS:[ECX+B8] | |
| 004650D3 | 8B00 | MOV EAX,DWORD PTR DS:[EAX] | |
| 004650D5 | 3D 940000C0 | CMP EAX,C0000094 | |
| 004650DA | 75 24 | JNZ SHORT 00465100 | UnPackMe.00465100 |
| 004650DC | FF81 B8000000 | INC DWORD PTR DS:[ECX+B8] | |
| 004650E2 | 33C0 | XOR EAX,EAX | |
| 004650E4 | 2141 04 | AND DWORD PTR DS:[ECX+4],EAX | |
| 004650E7 | 2141 08 | AND DWORD PTR DS:[ECX+8],EAX | |
| 004650EA | 2141 0C | AND DWORD PTR DS:[ECX+C],EAX | |
| 004650ED | 2141 10 | AND DWORD PTR DS:[ECX+10],EAX | |

这说明 ESP 定律不起作用,有可能该壳会检测硬件断点,我们继续运行的话会提示错误,所以我们暂时先把硬件断点删除掉。

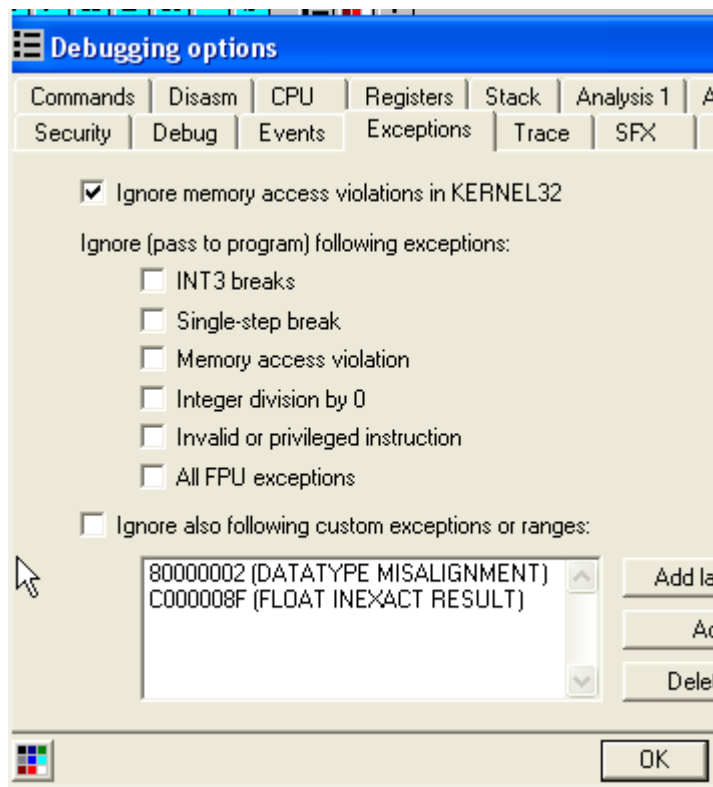


重启 OD。

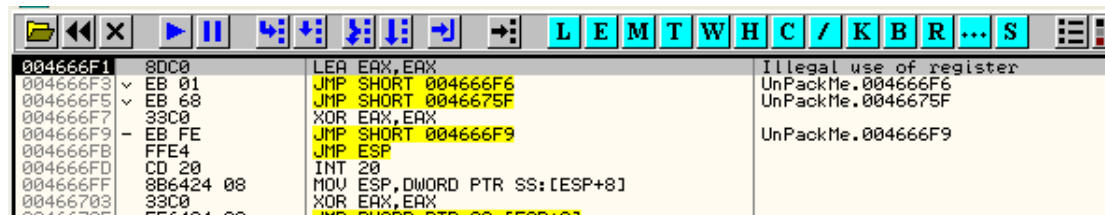
ESP 定律行不通,我们来尝试一下最后一次异常法,首先清空日志窗口。



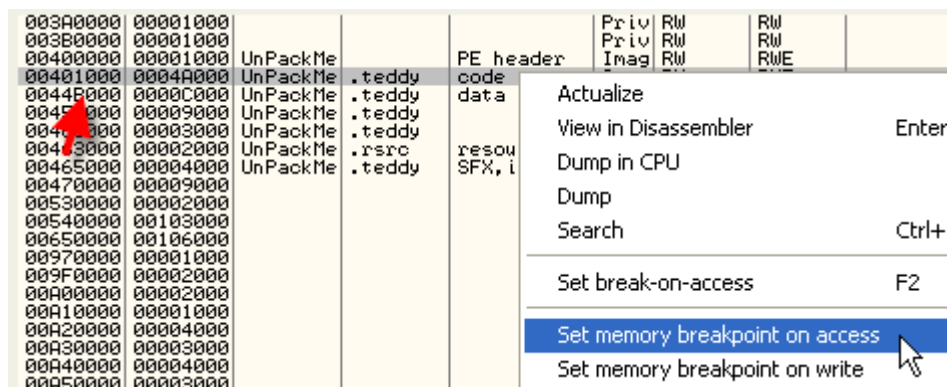
忽略异常的选项都不勾选,运行起来,可以看到壳的解密例程产生的最后一处异常是 4666F1 处。



运行起来,遇到异常直接 Shift + F9 忽略掉,直到断在 4666F1 处为止。



给第一个区段设置内存访问断点。



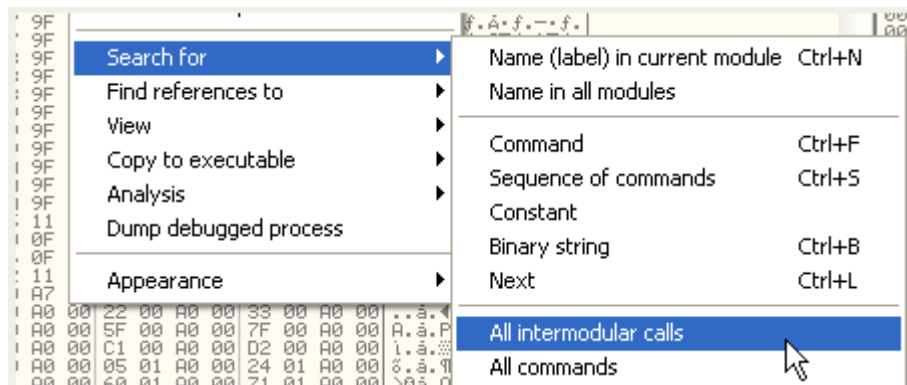
接着 Shift + F9 运行起来,遇到异常就 Shift + F9,不一会儿就会断在位于第一个区段中的 OEP 处。

| | | | |
|----------|------------------|--------------------------------|------------------------------|
| 00427180 | 55 | PUSH EBP | Real entry point of SFX code |
| 00427181 | 8BEC | MOV EBP, ESP | |
| 00427183 | 6A FF | PUSH -1 | |
| 00427185 | 68 600E4500 | PUSH 450E60 | |
| 0042718A | 68 C8924200 | PUSH 4292C8 | |
| 0042718F | 64:A1 00000000 | MOV EAX, DWORD PTR FS:[0] | |
| 004271C5 | 50 | PUSH EAX | |
| 004271C6 | 64:8925 00000000 | MOV DWORD PTR FS:[0], ESP | |
| 004271CD | 83C4 A8 | ADD ESP, -58 | |
| 004271D0 | 53 | PUSH EBX | |
| 004271D1 | 56 | PUSH ESI | |
| 004271D2 | 57 | PUSH EDI | |
| 004271D3 | 8965 E8 | MOV DWORD PTR SS:[EBP-18], ESP | |
| 004271D6 | FF15 DC0A4600 | CALL DWORD PTR DS:[460ADC] | |
| 004271DC | 33D2 | XOR EDX, EDX | |
| 004271DE | 80N4 | MOVI N1, 0H | |

因此 OEP 是 4271B0, 这里的原程序和上一个例子的原程序是一样的, 加的壳不一样而已。

我们知道上一例子原程序调用的第一个 API 函数是 GetVersion, 但是这个例子我们并没有看到 GetVersion 的提示信息, 被重定向向了。

我们来看看调用了哪些 API 函数, 在反汇编窗口中单击鼠标右键选择-Search for-All intermodular calls。



我们可以看到这些 CALL 都被重定向到了其他区段。

| Address | Disassembly | Destination |
|----------|----------------------------|-------------------------|
| 004231C8 | CALL FC4273FE | |
| 00423310 | CALL DWORD PTR DS:[460AB0] | DS:[00460AB0]=009F061C |
| 00423A4F | CALL DWORD PTR DS:[460A00] | DS:[00460A00]=009F02AF |
| 00423B1C | CALL DWORD PTR DS:[4609FC] | DS:[004609FC]=009F029E |
| 00423C43 | CALL DWORD PTR DS:[4609FC] | DS:[004609FC]=009F029E |
| 00423CA8 | CALL DWORD PTR DS:[4609F8] | DS:[004609F8]=009F028D |
| 00423E5C | CALL DWORD PTR DS:[460A58] | DS:[00460A58]=009F046A |
| 00423E96 | CALL DWORD PTR DS:[460B5C] | DS:[00460B5C]=009F0973 |
| 00424986 | CALL DWORD PTR DS:[4609F4] | DS:[004609F4]=009F027C |
| 00425003 | CALL DWORD PTR DS:[460B98] | DS:[00460B98]=009F0AA0 |
| 00425183 | CALL DWORD PTR DS:[460B98] | DS:[00460B98]=009F0AA0 |
| 004251C7 | CALL DWORD PTR DS:[460B94] | DS:[00460B94]=009F0A7D |
| 0042520B | CALL DWORD PTR DS:[460B94] | DS:[00460B94]=009F0A7D |
| 004252D4 | CALL DWORD PTR DS:[460B94] | DS:[00460B94]=009F0A7D |
| 004252FB | CALL DWORD PTR DS:[460978] | DS:[00460978]=009F0011 |
| 00425306 | CALL DWORD PTR DS:[460974] | DS:[00460974]=009F0000 |
| 0042535E | CALL DWORD PTR DS:[4609B0] | DS:[004609B0]=009F0124 |
| 004259D1 | CALL DWORD PTR DS:[460A54] | DS:[00460A54]=009F0447 |
| 004259E8 | CALL DWORD PTR DS:[460A3C] | DS:[00460A3C]=009F03DC |
| 004259F5 | CALL DWORD PTR DS:[460A44] | DS:[00460A44]=009F03FB |
| 00425B2B | CALL DWORD PTR DS:[460AF8] | DS:[00460AF8]=009F0785 |
| 00425B3F | CALL DWORD PTR DS:[460AF8] | DS:[00460AF8]=009F0785 |
| 00425B53 | CALL DWORD PTR DS:[460AF8] | DS:[00460AF8]=009F0785 |
| 00425BF1 | CALL DWORD PTR DS:[460AF8] | DS:[00460AF8]=009F0785 |
| 00425C9C | CALL DWORD PTR DS:[46097C] | DS:[0046097C]=009F0022 |
| 00425CA6 | CALL DWORD PTR DS:[460B5C] | DS:[00460B5C]=009F0973 |
| 00425D71 | CALL DWORD PTR DS:[460A54] | DS:[00460A54]=009F0447 |
| 00425D8B | CALL DWORD PTR DS:[460A3C] | DS:[00460A3C]=009F03DC |
| 00425E13 | CALL DWORD PTR DS:[460B98] | DS:[00460B98]=009F0AA0 |
| 00425E27 | CALL DWORD PTR DS:[460B94] | DS:[00460B94]=009F0A7D |
| 00425E6B | CALL DWORD PTR DS:[460B94] | DS:[00460B94]=009F0A7D |
| 00425F34 | CALL DWORD PTR DS:[460B94] | DS:[00460B94]=009F0A7D |
| 004271B0 | PUSH EBP | (Initial CPU selection) |
| 004271D6 | CALL DWORD PTR DS:[460ADC] | DS:[00460ADC]=009F06F7 |
| 0042723E | CALL DWORD PTR DS:[460984] | DS:[00460984]=009F0041 |
| 004272D5 | CALL DWORD PTR DS:[460980] | DS:[00460980]=009F0033 |
| 004272F6 | CALL DWORD PTR DS:[460B9C] | DS:[00460B9C]=009F0AB1 |
| 004274F7 | CALL 00435CC0 | UnPackMe.00435CC0 |
| 004277F3 | CALL 00435CC0 | UnPackMe.00435CC0 |
| 00427929 | CALL DWORD PTR DS:[46098C] | DS:[0046098C]=009F005F |
| 00427E07 | CALL DWORD PTR DS:[460A84] | DS:[00460A84]=009F0539 |
| 00427E0E | CALL DWORD PTR DS:[460994] | DS:[00460994]=009F008D |
| 00427E98 | CALL DWORD PTR DS:[460990] | DS:[00460990]=009F007F |
| 00428029 | CALL DWORD PTR DS:[4609F8] | DS:[004609F8]=009F028D |
| 004280B6 | CALL DWORD PTR DS:[460A08] | DS:[00460A08]=009F02CC |
| 00428823 | CALL DWORD PTR DS:[460A54] | DS:[00460A54]=009F0447 |
| 0042884D | CALL DWORD PTR DS:[460A3C] | DS:[00460A3C]=009F03DC |
| 0042886E | CALL DWORD PTR DS:[460A44] | DS:[00460A44]=009F03FB |
| 004288AE | CALL DWORD PTR DS:[460A3C] | DS:[00460A3C]=009F03DC |
| 004288E0 | CALL DWORD PTR DS:[460A3C] | DS:[00460A3C]=009F03DC |
| 0042891E | CALL DWORD PTR DS:[460A44] | DS:[00460A44]=009F03FB |
| 00428950 | CALL DWORD PTR DS:[460A44] | DS:[00460A44]=009F03FB |
| 00428983 | CALL DWORD PTR DS:[460B98] | DS:[00460B98]=009F0AA0 |
| 00428C15 | CALL DWORD PTR DS:[460B9C] | DS:[00460B9C]=009F0AB1 |

这些间接 CALL 并不是去调用系统 DLL 的中 API 函数,而是转向了 9Fxxxx 这类地址的一个区段,这里在我的机器上是 9Fxxxx,大家的机器上不一定是这个地址。

如果大家往下看的话,会发现还是有一些直接调用 API 函数的,但是大部分还是 CALL 9Fxxxx 这类指令。

| | | | | |
|----------|------|-----------|-------------|---------------------------|
| 00435B3F | CALL | DWORD PTR | DS:[460B74] | DS:[00460B74]=009F09F0 |
| 00435D96 | CALL | DWORD PTR | DS:[460B74] | DS:[00460B74]=009F09F0 |
| 00435DF1 | CALL | DWORD PTR | DS:[460AB0] | DS:[00460AB0]=009F061C |
| 00435E06 | CALL | DWORD PTR | DS:[460AB4] | DS:[00460AB4]=009F0636 |
| 00435FA5 | CALL | 00435CDE | | comdlg32.PrintDlgA |
| 00436004 | CALL | DWORD PTR | DS:[460AD0] | DS:[00460AD0]=009F06B5 |
| 0043602E | CALL | DWORD PTR | DS:[460AD0] | DS:[00460AD0]=009F06B5 |
| 0043606C | CALL | DWORD PTR | DS:[460AD0] | DS:[00460AD0]=009F06B5 |
| 004360AB | CALL | DWORD PTR | DS:[460AD0] | DS:[00460AD0]=009F06B5 |
| 004360D1 | CALL | 00435CDE | | comdlg32.PrintDlgA |
| 00436118 | CALL | DWORD PTR | DS:[460878] | DS:[00460878]=00A10124 |
| 00436230 | CALL | DWORD PTR | DS:[460B74] | DS:[00460B74]=009F09F0 |
| 004362EC | CALL | DWORD PTR | DS:[460CAC] | DS:[00460CAC]=00A00360 |
| 00436315 | CALL | DWORD PTR | DS:[460CD4] | DS:[00460CD4]=00A00419 |
| 00436351 | CALL | 00435CEA | | comdlg32.GetOpenFileNameA |
| 00436358 | CALL | 00435CE4 | | comdlg32.GetSaveFileNameA |
| 00436374 | CALL | DWORD PTR | DS:[460DF4] | DS:[00460DF4]=00A009B7 |
| 00436382 | CALL | DWORD PTR | DS:[460CB0] | DS:[00460CB0]=00A00370 |
| 00436407 | CALL | DWORD PTR | DS:[460E78] | DS:[00460E78]=00A00C52 |
| 0043644E | CALL | DWORD PTR | DS:[460E78] | DS:[00460E78]=00A00C52 |
| 004366CA | CALL | DWORD PTR | DS:[460DFC] | DS:[00460DFC]=00A009F0 |
| 004368C5 | CALL | DWORD PTR | DS:[460D48] | DS:[00460D48]=00A00669 |
| 004368DB | CALL | DWORD PTR | DS:[460D48] | DS:[00460D48]=00A00669 |
| 004368F1 | CALL | DWORD PTR | DS:[460D48] | DS:[00460D48]=00A00669 |
| 00436907 | CALL | DWORD PTR | DS:[460D48] | DS:[00460D48]=00A00669 |
| 00436910 | CALL | DWORD PTR | DS:[460D48] | DS:[00460D48]=00A00669 |
| 00436933 | CALL | DWORD PTR | DS:[460D48] | DS:[00460D48]=00A00669 |
| 00436A6A | CALL | DWORD PTR | DS:[460E78] | DS:[00460E78]=00A00C52 |
| 00436BD0 | CALL | 00435CFC | | comdlg32.FindTextA |
| 00436BD7 | CALL | 00435CF6 | | comdlg32.ReplaceTextA |
| 00436CB1 | CALL | DWORD PTR | DS:[460AA8] | DS:[00460AA8]=009F05EC |
| 00436CC3 | CALL | DWORD PTR | DS:[460AAC] | DS:[00460AAC]=009F05FD |
| 0043799F | CALL | DWORD PTR | DS:[460D48] | DS:[00460D48]=00A00669 |

这里这些直接调用 API 函数的项我用湛蓝色标注出来了。

| | | | |
|----------|-------------|--------------------|---------------------------|
| 00435FA5 | E8 34F0FFFF | CALL 00435CDE | JMP to comdlg32.PrintDlgA |
| 00435FAA | 8BCE | MOV ECX,ESI | |
| 00435FAC | 8BF8 | MOV EDI, EAX | |
| 00435FAE | E8 6E4E0000 | CALL 0043AE21 | UnPackMe.0043AE21 |
| 00435FB3 | 5EFF | TEST EDI, EDI | |
| 00435FB5 | 74 04 | JS SHORT 00435FBB | |
| 00435FB7 | 8BC7 | MOV EAX, EDI | UnPackMe.00435FBB |
| 00435FB9 | EB 03 | JMP SHORT 00435FBE | |
| 00435FBB | 6A 02 | PUSH 2 | UnPackMe.00435FBE |
| 00435FBD | 58 | POP EAX | |
| 00435FBE | 5F | POP EDI | |
| 00435FBF | 5E | POP ESI | |
| 00435FC0 | C3 | RETN | |

这里我们定位到 435FA5 这处,这里是 CALL 435CDE,OD 提示是一个间接跳转到 API 函数的入口处,我们在这条指令上单击鼠标右键选择-Follow。

| | | | |
|----------|---------------|---------------------------|-------------------------------|
| 00435CDE | CC | INT3 | |
| 00435CBF | 90 | NOP | |
| 00435CC0 | FF25 88094600 | JMP DWORD PTR DS:[460988] | |
| 00435CC6 | FF25 000C4600 | JMP DWORD PTR DS:[460C00] | |
| 00435CCC | FF25 040C4600 | JMP DWORD PTR DS:[460C04] | |
| 00435CD2 | FF25 000E4600 | JMP DWORD PTR DS:[460E00] | comdlg32.ChooseFontA |
| 00435CD8 | FF25 080E4600 | JMP DWORD PTR DS:[460E08] | comdlg32.ChooseColorA |
| 00435CDE | FF25 040E4600 | JMP DWORD PTR DS:[460E04] | comdlg32.PrintDlgA |
| 00435CE4 | FF25 C00E4600 | JMP DWORD PTR DS:[460EC0] | comdlg32.GetSaveFileNameA |
| 00435CEA | FF25 C80E4600 | JMP DWORD PTR DS:[460EC8] | comdlg32.GetOpenFileNameA |
| 00435CE8 | FF25 C40E4600 | JMP DWORD PTR DS:[460EC4] | comdlg32.GetFileTitleA |
| 00435CF6 | FF25 C00E4600 | JMP DWORD PTR DS:[460EC0] | comdlg32.ReplaceTextA |
| 00435CFC | FF25 BC0E4600 | JMP DWORD PTR DS:[460EBC] | comdlg32.FindTextA |
| 00435D02 | FF25 B80E4600 | JMP DWORD PTR DS:[460EB8] | comdlg32.CommDlgExtendedError |
| 00435D08 | FF25 B00E4600 | JMP DWORD PTR DS:[460EB0] | WINSPool.ClosePrinter |
| 00435D0E | FF25 940E4600 | JMP DWORD PTR DS:[460E94] | WINSPool.EndDocPrinter |
| 00435D14 | FF25 AC0E4600 | JMP DWORD PTR DS:[460EAC] | WINSPool.StartPagePrinter |
| 00435D1A | FF25 A80E4600 | JMP DWORD PTR DS:[460EA8] | WINSPool.StartDocPrinterA |
| 00435D20 | FF25 A40E4600 | JMP DWORD PTR DS:[460EA4] | WINSPool.OpenPrinterA |
| 00435D26 | FF25 A00E4600 | JMP DWORD PTR DS:[460EA0] | WINSPool.EndPagePrinter |
| 00435D2C | FF25 9C0E4600 | JMP DWORD PTR DS:[460E9C] | WINSPool.WritePrinter |
| 00435D32 | FF25 980E4600 | JMP DWORD PTR DS:[460E98] | WINSPool.DocumentPropertiesA |
| 00435D38 | FF25 240F4600 | JMP DWORD PTR DS:[460F24] | oledlg.OleUIBusyA |
| 00435D3E | CC | INT3 | |
| 00435D40 | C3 | RETN | |

我们可以看到是一些间接跳转去调用 API 函数,很明显这里是 IAT 中的一些项,我们在数据窗口中定位到 460ED4 这一项。

| Address | Hex dump | ASCII |
|----------|---|-------------------|
| 00460EB4 | 00 00 00 00 CE 00 37 76 7C 86 37 76 B0 86 37 76 | ...if.7v!37v37v |
| 00460EC4 | 33 25 36 76 1E 31 36 76 D8 7C 37 76 89 C2 37 76 | 3%6v!16v!17v37v |
| 00460ED4 | CD 46 38 76 CE EE 36 76 00 00 00 00 48 D0 4C 77 | =F8v!6v...H\$Lw |
| 00460EE4 | 9C CB 4D 77 CC 42 4F 77 2C D0 4C 77 DA F6 4C 77 | 5fHw!fB0w,\$Lw!Lw |
| 00460EF4 | 73 33 50 77 10 64 4D 77 03 0E 52 77 33 0F 52 77 | s3Pw!dHw!8Rw3*Rw |
| 00460F04 | 40 A6 54 77 F1 A7 54 77 92 9C 4F 77 6F 57 52 77 | @3Tw!2Tw!E60w!Rw |
| 00460F14 | 99 33 4E 77 B2 5D 4E 77 90 C0 5A 77 00 00 00 00 | 03Nw!Nw!2w.... |
| 00460F24 | F3 F0 CC 74 00 00 00 00 00 00 00 00 00 00 00 | %-!ft..... |
| 00460F34 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | |
| 00460F44 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | |
| 00460F54 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | |
| 00460F64 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | |
| 00460F74 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | |
| 00460F84 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | |
| 00460F94 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | |
| 00460FA4 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | |

这里我们可以看到 IAT 中的最后一部分是正确的,跟上一个例子一样,IAT 的结束地址为 460F28,下面全是零了。

| Address | Hex dump | ASCII |
|----------|---|--|
| 00460D54 | 95 06 A0 00 B5 06 A0 00 C3 06 A0 00 E6 06 A0 00 | 00 |
| 00460D64 | F7 06 A0 00 08 07 A0 00 1A 07 A0 00 2A 07 A0 00 | 00 |
| 00460D74 | 3B 07 A0 00 5A 07 A0 00 74 07 A0 00 85 07 A0 00 | 00 |
| 00460D84 | 96 07 A0 00 A7 07 A0 00 B5 07 A0 00 C4 07 A0 00 | 00 |
| 00460D94 | 03 07 A0 00 F3 07 A0 00 01 08 A0 00 24 08 A0 00 | 00 |
| 00460DA4 | 35 08 A0 00 46 08 A0 00 58 08 A0 00 68 08 A0 00 | 00 |
| 00460DB4 | 79 08 A0 00 98 08 A0 00 B2 08 A0 00 C3 08 A0 00 | 00 |
| 00460DC4 | 04 08 A0 00 E5 08 A0 00 F3 08 A0 00 02 09 A0 00 | 00 |
| 00460DD4 | 11 09 A0 00 31 09 A0 00 3F 09 A0 00 62 09 A0 00 | 00 |
| 00460DE4 | 73 09 A0 00 84 09 A0 00 96 09 A0 00 A6 09 A0 00 | 00 |
| 00460DF4 | B7 09 A0 00 D6 09 A0 00 F0 09 A0 00 01 0A 00 00 | 00 |
| 00460E04 | 12 0A 00 00 23 0A 00 00 31 0A 00 00 40 0A 00 00 | 00 |
| 00460E14 | 4F 0A 00 00 6F 0A 00 00 7D 0A 00 00 A0 0A 00 00 | 00 |
| 00460E24 | B1 0A 00 00 C2 0A 00 00 D4 0A 00 00 E4 0A 00 00 | 00 |
| 00460E34 | F5 0A 00 00 14 0B A0 00 2E 0B A0 00 3F 0B A0 00 | 00 |
| 00460E44 | 50 0B A0 00 61 0B A0 00 6F 0B A0 00 7E 0B A0 00 | 00 |
| 00460E54 | 8D 0B A0 00 AD 0B A0 00 BB 0B A0 00 DE 0B A0 00 | 00 |
| 00460E64 | EF 0B A0 00 00 0C A0 00 12 0C A0 00 22 0C A0 00 | 00 |
| 00460E74 | 33 0C A0 00 52 0C A0 00 6C 0C A0 00 7D 0C A0 00 | 00 |
| 00460E84 | 8E 0C A0 00 9F 0C A0 00 F7 A8 B1 76 00 00 00 00 | 00 |
| 00460E94 | C8 74 F8 72 73 66 F9 72 87 72 F8 72 43 80 F8 72 | 00 |
| 00460EA4 | 67 37 F9 72 FB 41 F9 72 67 83 F8 72 90 53 F8 72 | 00 |
| 00460EB4 | 00 00 00 00 CE 00 37 76 7C 86 37 76 B0 86 37 76 | ...if.7v!37v37v |
| 00460EC4 | 33 25 36 76 1E 31 36 76 D8 7C 37 76 89 C2 37 76 | 3%6v!16v!17v37v |
| 00460ED4 | CD 46 38 76 CE EE 36 76 00 00 00 00 48 D0 4C 77 | =F8v!6v...H\$Lw |
| 00460EE4 | 9C CB 4D 77 CC 42 4F 77 2C D0 4C 77 DA F6 4C 77 | 5fHw!fB0w,\$Lw!Lw |
| 00460EF4 | 73 33 50 77 10 64 4D 77 03 0E 52 77 33 0F 52 77 | s3Pw!dHw!8Rw3*Rw |
| 00460F04 | 40 A6 54 77 F1 A7 54 77 92 9C 4F 77 6F 57 52 77 | @3Tw!2Tw!E60w!Rw |
| 00460F14 | 99 33 4E 77 B2 5D 4E 77 90 C0 5A 77 00 00 00 00 | 03Nw!Nw!2w.... |
| 00460F24 | F3 F0 CC 74 00 00 00 00 00 00 00 00 00 00 00 | %-!ft..... |
| 00460F34 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | |
| 00460F44 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | |
| 00460F54 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | |
| 00460F64 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | |

往上看,这里用湛蓝色标注出来的这一部分,这一组里面的其他项形式跟 76B1A8F7 这一项形式不一样,我们在 76B1A8F7 这一项上面单击鼠标右键选择-Find references。

| Address | Disassembly | Comment |
|----------|----------------------------|-------------------|
| 0040E203 | CALL DWORD PTR DS:[460E8C] | WINMM.PlaySoundA |
| 00460E95 | JE SHORT 00460E8F | UnPackMe.00460E8F |

我们可以看到调用的是 WINMM.dll 中的 PlaySoundA 这个 API 函数,我们再来看看其他项的参考引用。

| Address | Disassembly | Comment |
|----------|----------------------------|------------------------|
| 004038A6 | CALL DWORD PTR DS:[460E48] | DS:[00460E48]=00A00B61 |
| 004047D0 | CALL DWORD PTR DS:[460E48] | DS:[00460E48]=00A00B61 |
| 00404923 | CALL DWORD PTR DS:[460E48] | DS:[00460E48]=00A00B61 |
| 00404AD9 | CALL DWORD PTR DS:[460E48] | DS:[00460E48]=00A00B61 |
| 0041331B | CALL DWORD PTR DS:[460E48] | DS:[00460E48]=00A00B61 |
| 004171DE | CALL DWORD PTR DS:[460E48] | DS:[00460E48]=00A00B61 |
| 0041A61E | CALL DWORD PTR DS:[460E48] | DS:[00460E48]=00A00B61 |
| 00421045 | CALL DWORD PTR DS:[460E48] | DS:[00460E48]=00A00B61 |
| 0043D912 | CALL DWORD PTR DS:[460E48] | DS:[00460E48]=00A00B61 |
| 0043DCC0 | CALL DWORD PTR DS:[460E48] | DS:[00460E48]=00A00B61 |
| 0043E9D6 | CALL DWORD PTR DS:[460E48] | DS:[00460E48]=00A00B61 |
| 0043EA8A | CALL DWORD PTR DS:[460E48] | DS:[00460E48]=00A00B61 |
| 00441B0A | CALL DWORD PTR DS:[460E48] | DS:[00460E48]=00A00B61 |

这里我们可以看到参考引用的一些项,但是都是一些指针,并不是调用某个系统 DLL 中的 API 函数,我这里是 Axxxxx 的形式(换一

台机器这类数值可能不一样),这类数值是什么呢?

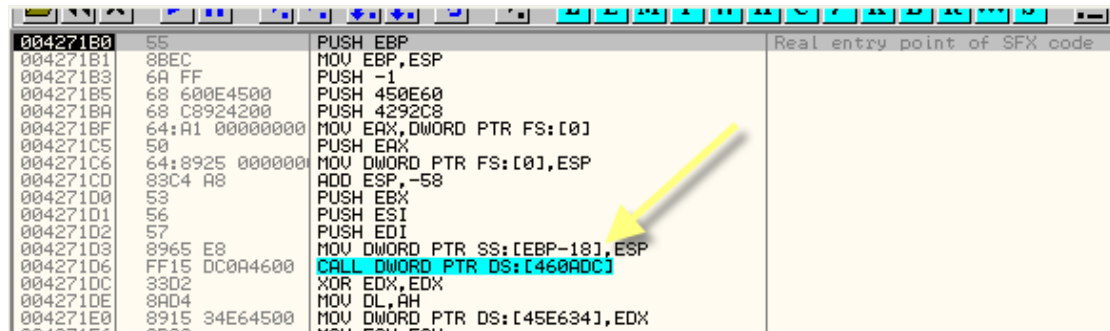
这些就是 IAT 中重定向的一些项,当程序运行起来时,壳的解密例会覆盖掉 IAT 中的某些项,将这些项重定向到解密例程中,我们拿 4038A6 此处为例:

```
004038A6 CALL DWORD PTR DS:[460E48]
```

```
Comment=DS:[00460E48]=00A00B61
```

这里并不是直接调用 API 函数,壳将这一项覆盖为了自己所在区段中的一个地址。

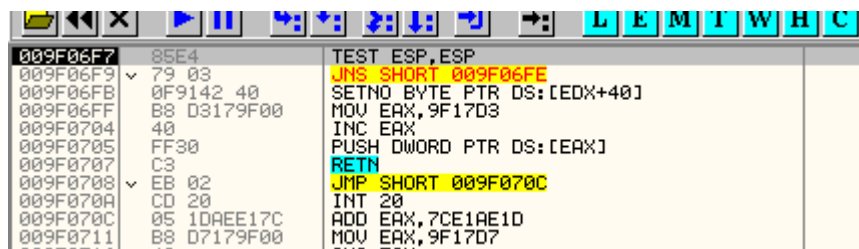
为了更好的理解这一点,我们来看看 GetVersion 这个 API 函数,在 OEP 的下面。



其实,这里原本我们并不知道是调用的 GetVersion,但是该原程序跟上一个例子是相同的,所以可以断定这里是调用 GetVersion,这里是一个间接 CALL,我们按 F7 键跟进,看看里面是什么。

| Address | Hex dump | ASCII |
|----------|---|-----------------|
| 00460A9C | B9 05 9F 00 CA 05 9F 00 DC 05 9F 00 EC 05 9F 00 | ! |
| 00460AAC | FD 05 9F 00 1C 06 9F 00 36 06 9F 00 47 06 9F 00 | 2 |
| 00460ABC | 58 06 9F 00 69 06 9F 00 77 06 9F 00 86 06 9F 00 | X |
| 00460ACC | 95 06 9F 00 B5 06 9F 00 C3 06 9F 00 E6 06 9F 00 | o |
| 00460ADC | F7 06 9F 00 08 07 9F 00 1A 07 9F 00 2A 07 9F 00 | = |
| 00460AEC | 3B 07 9F 00 5A 07 9F 00 74 07 9F 00 85 07 9F 00 | ; |
| 00460AFC | 96 07 9F 00 A7 07 9F 00 B5 07 9F 00 C4 07 9F 00 | U |
| 00460B0C | D3 07 9F 00 F3 07 9F 00 01 08 9F 00 24 08 9F 00 | E |
| 00460B1C | 35 08 9F 00 46 08 9F 00 58 08 9F 00 68 08 9F 00 | S |
| 00460B2C | 79 08 9F 00 98 08 9F 00 B2 08 9F 00 C3 08 9F 00 | y |
| 00460B3C | D4 08 9F 00 E5 08 9F 00 F3 08 9F 00 02 09 9F 00 | e |
| 00460B4C | 11 09 9F 00 31 09 9F 00 3F 09 9F 00 62 09 9F 00 | l |
| 00460B5C | 73 09 9F 00 84 09 9F 00 96 09 9F 00 A6 09 9F 00 | s |
| 00460B6C | B7 09 9F 00 D6 09 9F 00 F0 09 9F 00 01 0A 9F 00 | A |
| 00460B7C | 12 0A 9F 00 23 0A 9F 00 31 0A 9F 00 40 0A 9F 00 | # |
| 00460B8C | 4F 0A 9F 00 6F 0A 9F 00 7D 0A 9F 00 A0 0A 9F 00 | O |
| 00460B9C | B1 0A 9F 00 C2 0A 9F 00 D4 0A 9F 00 C0 4B 0F 77 | W |
| 00460BAC | 38 4C 0F 77 94 A5 11 77 59 4B 0F 77 82 4E 0F 77 | L ; |
| 00460BBC | 98 04 11 77 08 50 0F 77 4F 50 0F 77 10 50 0F 77 | U ; |
| 00460BCC | 3F 50 0F 77 09 66 0F 77 50 4B 0F 77 55 4C 0F 77 | ? P ; |
| 00460BDC | C2 4B 0F 77 95 02 11 77 80 5D 15 77 00 00 00 00 | T K ; |
| 00460BEC | 00 00 A7 00 11 00 A7 00 22 00 A7 00 33 00 A7 00 | |
| 00460BFC | 00 00 A0 00 11 00 A0 00 22 00 A0 00 33 00 A0 00 | |
| 00460C0C | 41 00 A0 00 50 00 A0 00 5F 00 A0 00 7F 00 A0 00 | A |
| 00460C1C | 80 00 A0 00 B0 00 A0 00 C1 00 A0 00 D2 00 A0 00 | i |
| 00460C2C | E4 00 A0 00 F4 00 A0 00 05 01 A0 00 24 01 A0 00 | % |
| 00460C3C | 3E 01 A0 00 4F 01 A0 00 60 01 A0 00 71 01 A0 00 | > @ |
| 00460C4C | 7F 01 A0 00 8E 01 A0 00 9D 01 A0 00 BD 01 A0 00 | @ @ |
| 00460C5C | CB 01 A0 00 EE 01 A0 00 FF 01 A0 00 10 02 A0 00 | ~ @ |
| 00460C6C | 22 02 A0 00 32 02 A0 00 43 02 A0 00 62 02 A0 00 | " @ |
| 00460C7C | 7C 02 A0 00 8D 02 A0 00 9E 02 A0 00 AF 02 A0 00 | ! @ |
| 00460C8C | BD 02 A0 00 CC 02 A0 00 DB 02 A0 00 FB 02 A0 00 | c @ |
| 00460C9C | 09 03 A0 00 2C 03 A0 00 3D 03 A0 00 4E 03 A0 00 | . @ |
| 00460CAC | 60 03 A0 00 70 03 A0 00 81 03 A0 00 A0 03 A0 00 | ' @ |
| 00460CBC | BA 03 A0 00 CB 03 A0 00 DC 03 A0 00 ED 03 A0 00 | @ |
| 00460CC | FB 03 A0 00 0A 04 A0 00 19 04 A0 00 39 04 A0 00 | ' @ |
| 00460CDC | 47 04 A0 00 6A 04 A0 00 7B 04 A0 00 8C 04 A0 00 | G @ |
| 00460CEC | 9E 04 A0 00 AE 04 A0 00 BF 04 A0 00 DE 04 A0 00 | x @ |
| 00460CF | F8 04 A0 00 09 05 A0 00 10 05 A0 00 2B 05 A0 00 | o @ |

我们到了 9F06F7 处。



这里我的机器上是 9F06F7,大家的机器上这个地址可能会不一样,该地址不属于原程序的区段。

| | | | | | | | | |
|----------|----------|----------|--------|--------------|------|-----|-----|----------------------|
| 00260000 | 00003000 | | | | Map | RW | RW | |
| 00270000 | 00016000 | | | | Map | R | R | \Device\HarddiskVolu |
| 00290000 | 0003D000 | | | | Map | R | R | \Device\HarddiskVolu |
| 002D0000 | 00041000 | | | | Map | R | R | \Device\HarddiskVolu |
| 00320000 | 00006000 | | | | Map | R | R | \Device\HarddiskVolu |
| 00330000 | 00041000 | | | | Map | R | R | |
| 00380000 | 00001000 | | | | Priv | RWE | RWE | |
| 00390000 | 00001000 | | | | Priv | RWE | RWE | |
| 003A0000 | 00001000 | | | | Priv | RW | RW | |
| 003B0000 | 00001000 | | | | Priv | RW | RW | |
| 00400000 | 00001000 | UnPackMe | | PE header | Imag | RW | RWE | |
| 00401000 | 0004A000 | UnPackMe | .teddy | code | Imag | RW | RWE | |
| 0044B000 | 0000C000 | UnPackMe | .teddy | data | Imag | RW | RWE | |
| 00457000 | 00009000 | UnPackMe | .teddy | | Imag | RW | RWE | |
| 00460000 | 00003000 | UnPackMe | .teddy | | Imag | RW | RWE | |
| 00463000 | 00002000 | UnPackMe | .rsrc | resources | Imag | RW | RWE | |
| 00465000 | 00004000 | UnPackMe | .teddy | SFX, imports | Imag | RW | RWE | |
| 00470000 | 00009000 | | | | Map | R E | R E | |
| 00530000 | 00002000 | | | | Map | R E | R E | |
| 00540000 | 00103000 | | | | Map | R | R | |
| 00650000 | 00106000 | | | | Map | R E | R E | |
| 00970000 | 00001000 | | | | Priv | RW | RW | |
| 009F0000 | 00002000 | | | | Priv | RW | RW | |
| 00A00000 | 00002000 | | | | Priv | RW | RW | |
| 00A10000 | 00001000 | | | | Priv | RW | RW | |
| 00A20000 | 00004000 | | | | Priv | RW | RW | |
| 00A30000 | 00003000 | | | | Map | R | R | \Device\HarddiskVolu |
| 00A40000 | 00004000 | | | | Priv | RW | RW | |
| 00A50000 | 00003000 | | | | Priv | RW | RW | |
| 00A60000 | 00002000 | | | | Map | R | R | |
| 00A70000 | 00001000 | | | | Priv | RW | RW | |
| A1270000 | 00002000 | | | | Map | R | R | |

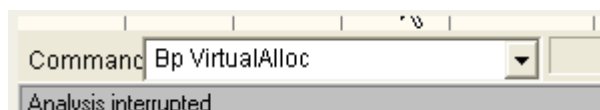
这里我们可以看到区段列表窗口,湛蓝色标注出来的是原程序区段,9F06F7 命中在下面这个没有标注名称起始地址为 9F0000 的区段。

如果我们重启 OD 断在入口点处时,会发现这个时候该区段并不存在。

| | | | | | | | | |
|----------|----------|----------|--------|--------------|------|-----|-----|--|
| 00330000 | 00041000 | | | | Map | R | R | |
| 00380000 | 00001000 | | | | Priv | RWE | RWE | |
| 00390000 | 00001000 | | | | Priv | RWE | RWE | |
| 003A0000 | 00001000 | | | | Priv | RW | RW | |
| 003B0000 | 00001000 | | | | Priv | RW | RW | |
| 00400000 | 00001000 | UnPackMe | | PE header | Imag | R | RWE | |
| 00401000 | 0004A000 | UnPackMe | .teddy | code | Imag | R | RWE | |
| 0044B000 | 0000C000 | UnPackMe | .teddy | data | Imag | R | RWE | |
| 00457000 | 00009000 | UnPackMe | .teddy | | Imag | R | RWE | |
| 00460000 | 00003000 | UnPackMe | .teddy | | Imag | R | RWE | |
| 00463000 | 00002000 | UnPackMe | .rsrc | resources | Imag | R | RWE | |
| 00465000 | 00004000 | UnPackMe | .teddy | SFX, imports | Imag | R | RWE | |
| 00470000 | 00009000 | | | | Map | R E | R E | |
| 00530000 | 00002000 | | | | Map | R E | R E | |
| 00540000 | 00103000 | | | | Map | R | R | |
| 00650000 | 00106000 | | | | Map | R E | R E | |
| 77D10000 | 00001000 | user32 | | PE header | Imag | R | RWE | |
| 77D11000 | 0005F000 | user32 | .text | code, import | Imag | R | RWE | |
| 77D70000 | 00002000 | user32 | .data | data | Imag | R | RWE | |
| 77D72000 | 00002000 | user32 | .rsrc | resources | Imag | R | RWE | |

因此,这个内存块是在壳解密例程运行过程中创建的,我们来看看它是什么时候被创建的。

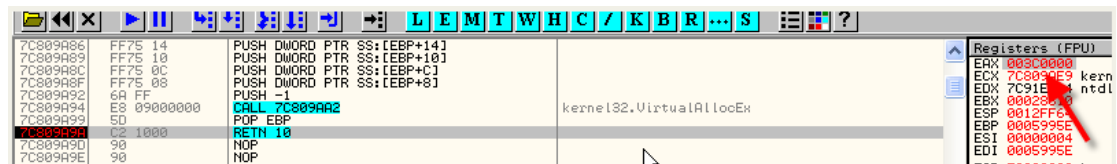
我们给 VirtualAlloc 这个 API 函数设置一个断点,这个函数是用来申请内存空间的。



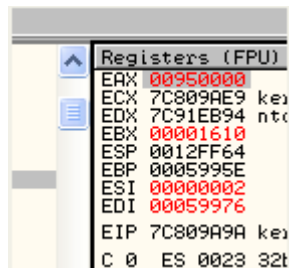
忽略异常的选项都勾选上,按 shift + F9 运行起来,程序直接结束了,很明显该壳会检测我们设置的 API 断点,我们将断点设置 VirtualAlloc 的 RET 处。

| | | | |
|----------|-------------|----------------------------|-------------------------|
| 7C809A86 | FF75 14 | PUSH DWORD PTR SS:[EBP+14] | |
| 7C809A89 | FF75 10 | PUSH DWORD PTR SS:[EBP+10] | |
| 7C809A8C | FF75 0C | PUSH DWORD PTR SS:[EBP+C] | |
| 7C809A8F | FF75 08 | PUSH DWORD PTR SS:[EBP+8] | |
| 7C809A92 | 6A FF | PUSH -1 | |
| 7C809A94 | E8 09000000 | CALL 7C809AA2 | kernel32.VirtualAllocEx |
| 7C809A99 | 5D | POP EBP | |
| 7C809A9A | C2 1000 | RETN 10 | |
| 7C809A9D | 90 | NOP | |
| 7C809A9E | 90 | NOP | |
| 7C809A9F | 90 | NOP | |

运行起来。



断了下来,EAX 的值为 3C0000,继续运行。

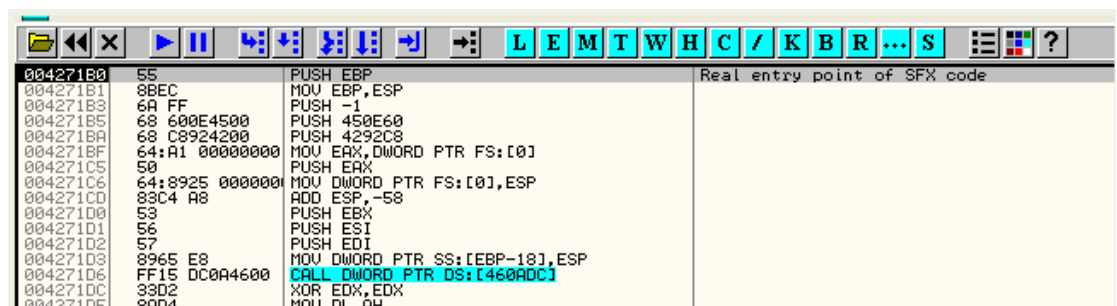


这里我们可以在区段列表窗口中看看刚刚创建的这个内存块。

| | | | | | | | | |
|----------|----------|----------|--------|--------------|------|-----|-----|--|
| 003B0000 | 00001000 | | | | Priv | RW | RW | |
| 00400000 | 00001000 | UnPackMe | | PE header | Imag | R | RWE | |
| 00401000 | 00004000 | UnPackMe | .teddy | code | Imag | R | RWE | |
| 0044B000 | 0000C000 | UnPackMe | .teddy | data | Imag | R | RWE | |
| 00457000 | 00009000 | UnPackMe | .teddy | | Imag | R | RWE | |
| 00460000 | 00003000 | UnPackMe | .teddy | | Imag | R | RWE | |
| 00463000 | 00002000 | UnPackMe | .rsrc | resources | Imag | R | RWE | |
| 00465000 | 00004000 | UnPackMe | .teddy | SFX, imports | Imag | R | RWE | |
| 00470000 | 00009000 | | | | Map | R E | R E | |
| 00530000 | 00002000 | | | | Map | R E | R E | |
| 00540000 | 00103000 | | | | Map | R | R | |
| 00650000 | 00106000 | | | | Map | R E | R E | |
| 00950000 | 00002000 | | | | Priv | RW | RW | |

我们可以看到该内存块被标记为了 Priv(私有的),也就是说该内存块是壳自己创建的。

这里我们删除之前设置的断点,按前面介绍的步骤利用最后一次异常法再次定位到 OEP 处。



我们再次定位到 OEP 处,来看看区段列表窗口。

| | | | | | | | | |
|----------|----------|----------|--------|--------------|------|-----|-----|--|
| 00400000 | 00001000 | UnPackMe | | PE header | Imag | RW | RWE | |
| 00401000 | 00004000 | UnPackMe | .teddy | code | Imag | RW | RWE | |
| 0044B000 | 0000C000 | UnPackMe | .teddy | data | Imag | RW | RWE | |
| 00457000 | 00009000 | UnPackMe | .teddy | | Imag | RW | RWE | |
| 00460000 | 00003000 | UnPackMe | .teddy | | Imag | RW | RWE | |
| 00463000 | 00002000 | UnPackMe | .rsrc | resources | Imag | RW | RWE | |
| 00465000 | 00004000 | UnPackMe | .teddy | SFX, imports | Imag | RW | RWE | |
| 00470000 | 00009000 | | | | Map | R E | R E | |
| 00530000 | 00002000 | | | | Map | R E | R E | |
| 00540000 | 00103000 | | | | Map | R | R | |
| 00650000 | 00106000 | | | | Map | R E | R E | |
| 00970000 | 00001000 | | | | Priv | RW | RW | |
| 009F0000 | 00002000 | | | | Priv | RW | RW | |
| 00A00000 | 00002000 | | | | Priv | RW | RW | |
| 00A10000 | 00001000 | | | | Priv | RW | RW | |
| 00A20000 | 00004000 | | | | Priv | RW | RW | |
| 00A30000 | 00003000 | | | | Map | R | R | |
| 00A40000 | 00004000 | | | | Priv | RW | RW | |
| 00A50000 | 00003000 | | | | Priv | RW | RW | |
| 00A60000 | 00002000 | | | | Map | R | R | |
| 00A70000 | 00001000 | | | | Priv | RW | RW | |
| 01270000 | 00002000 | | | | Map | R | R | |
| 58C30000 | 00001000 | COMCTL32 | | PE header | Imag | R | RWE | |
| 58C31000 | 00070000 | COMCTL32 | .text | code, import | Imag | R | RWE | |
| 58CA1000 | 00003000 | COMCTL32 | .data | data | Imag | R | RWE | |

可能看到起始地址为 9F0000 这块内存空间也被标记为了 Priv,IAT 项中的值会被重定向到这里,我们按 F7 键单步到 9F06F7 处。

| | | |
|----------|-------------|----------------------------|
| 009F06F7 | 85E4 | TEST ESP,ESP |
| 009F06F9 | 79 03 | JNS SHORT 009F06FE |
| 009F06FB | 0F9142 40 | SETNO BYTE PTR DS:[EDX+40] |
| 009F06FF | B8 D3179F00 | MOV EAX,9F17D3 |
| 009F0704 | 40 | INC EAX |
| 009F0705 | FF30 | PUSH DWORD PTR DS:[EAX] |
| 009F0707 | C3 | RETN |
| 009F0708 | EB 02 | JMP SHORT 009F070C |
| 009F070A | CD 20 | INT 20 |
| 009F070C | 05 1DAEE17C | ADD EAX,7CE1AE1D |
| 009F0711 | B8 D7179F00 | MOV EAX,9F17D7 |

定位到了 9F06F7 处。

| | | | |
|----------|-------------|-------------------------|---------------------|
| 009F06FE | 40 | INC EAX | |
| 009F06FF | B8 D3179F00 | MOV EAX,9F17D3 | |
| 009F0704 | 40 | INC EAX | |
| 009F0705 | FF30 | PUSH DWORD PTR DS:[EAX] | kernel32.GetVersion |
| 009F0707 | C3 | RETN | |
| 009F0708 | EB 02 | JMP SHORT 009F070C | |
| 009F070A | CD 20 | INT 20 | |
| 009F070C | 05 1DAEE17C | ADD EAX,7CE1AE1D | |
| 009F0711 | B8 D7179F00 | MOV EAX,9F17D7 | |

往下跟几步,可以看到这里的 PUSH 指令将 GetVersion 的地址压入到堆栈中,接着 RET 将会返回到 GetVersion 的入口处,这样就可以达到间接调用 API 函数的目的。

也就是说,壳会将 GetVersion 的 IAT 项替换成自己创建的内存单元中的地址,起到了一个重定向的作用。

因此我们在定位 IAT 的起始和结束位置的时候,不仅仅要判断是否为系统 DLL 中的地址,还是需要判断其是否为重定向过的地址。

下面我们继续来定位 IAT 的起始和结束位置。

| Address | Hex dump | ASCII |
|----------|---|----------------------------------|
| 00460C5C | CB 01 A0 00 EE 01 A0 00 FF 01 A0 00 10 02 A0 00 | 00000000000000000000000000000000 |
| 00460C6C | 22 02 A0 00 32 02 A0 00 43 02 A0 00 62 02 A0 00 | 00000000000000000000000000000000 |
| 00460C7C | 7C 02 A0 00 80 02 A0 00 9E 02 A0 00 AF 02 A0 00 | 00000000000000000000000000000000 |
| 00460C8C | BD 02 A0 00 CC 02 A0 00 DB 02 A0 00 FB 02 A0 00 | 00000000000000000000000000000000 |
| 00460C9C | 09 03 A0 00 2C 03 A0 00 3D 03 A0 00 4E 03 A0 00 | 00000000000000000000000000000000 |
| 00460CAC | 60 03 A0 00 70 03 A0 00 81 03 A0 00 A0 03 A0 00 | 00000000000000000000000000000000 |
| 00460CBC | BA 03 A0 00 CB 03 A0 00 DC 03 A0 00 ED 03 A0 00 | 00000000000000000000000000000000 |
| 00460CCC | FB 03 A0 00 0A 04 A0 00 19 04 A0 00 39 04 A0 00 | 00000000000000000000000000000000 |
| 00460CDC | 47 04 A0 00 6A 04 A0 00 7B 04 A0 00 8C 04 A0 00 | 00000000000000000000000000000000 |
| 00460CEC | 9E 04 A0 00 AE 04 A0 00 BF 04 A0 00 DE 04 A0 00 | 00000000000000000000000000000000 |
| 00460CFC | F8 04 A0 00 09 05 A0 00 1A 05 A0 00 2B 05 A0 00 | 00000000000000000000000000000000 |
| 00460D0C | 39 05 A0 00 48 05 A0 00 57 05 A0 00 77 05 A0 00 | 00000000000000000000000000000000 |
| 00460D1C | 85 05 A0 00 A8 05 A0 00 B9 05 A0 00 CA 05 A0 00 | 00000000000000000000000000000000 |
| 00460D2C | DC 05 A0 00 EC 05 A0 00 FD 05 A0 00 1C 06 A0 00 | 00000000000000000000000000000000 |
| 00460D3C | 36 06 A0 00 47 06 A0 00 58 06 A0 00 69 06 A0 00 | 00000000000000000000000000000000 |
| 00460D4C | 77 06 A0 00 86 06 A0 00 95 06 A0 00 B5 06 A0 00 | 00000000000000000000000000000000 |
| 00460D5C | C3 06 A0 00 E6 06 A0 00 F7 06 A0 00 08 07 A0 00 | 00000000000000000000000000000000 |
| 00460D6C | 1A 07 A0 00 2A 07 A0 00 3B 07 A0 00 5A 07 A0 00 | 00000000000000000000000000000000 |
| 00460D7C | 74 07 A0 00 85 07 A0 00 96 07 A0 00 A7 07 A0 00 | 00000000000000000000000000000000 |
| 00460D8C | B5 07 A0 00 C4 07 A0 00 D3 07 A0 00 F3 07 A0 00 | 00000000000000000000000000000000 |
| 00460D9C | 01 08 A0 00 24 08 A0 00 35 08 A0 00 46 08 A0 00 | 00000000000000000000000000000000 |
| 00460DAC | 58 08 A0 00 68 08 A0 00 79 08 A0 00 98 08 A0 00 | 00000000000000000000000000000000 |
| 00460DBC | B2 08 A0 00 C3 08 A0 00 D4 08 A0 00 E5 08 A0 00 | 00000000000000000000000000000000 |
| 00460DCC | F3 08 A0 00 02 09 A0 00 11 09 A0 00 31 09 A0 00 | 00000000000000000000000000000000 |
| 00460DDC | 3F 09 A0 00 62 09 A0 00 73 09 A0 00 84 09 A0 00 | 00000000000000000000000000000000 |
| 00460DEC | 96 09 A0 00 A6 09 A0 00 B7 09 A0 00 D6 09 A0 00 | 00000000000000000000000000000000 |
| 00460DFC | F0 09 A0 00 01 0A A0 00 12 0A A0 00 23 0A A0 00 | 00000000000000000000000000000000 |
| 00460E0C | 31 0A A0 00 40 0A A0 00 4F 0A A0 00 6F 0A A0 00 | 00000000000000000000000000000000 |
| 00460E1C | 7D 0A A0 00 A0 0A A0 00 B1 0A A0 00 C2 0A A0 00 | 00000000000000000000000000000000 |
| 00460E2C | 04 0A A0 00 E4 0A A0 00 F5 0A A0 00 14 0B A0 00 | 00000000000000000000000000000000 |
| 00460E3C | 2E 0B A0 00 3F 0B A0 00 50 0B A0 00 61 0B A0 00 | 00000000000000000000000000000000 |
| 00460E4C | 6F 0B A0 00 7E 0B A0 00 8D 0B A0 00 AD 0B A0 00 | 00000000000000000000000000000000 |
| 00460E5C | B8 0B A0 00 DE 0B A0 00 EF 0B A0 00 00 0C A0 00 | 00000000000000000000000000000000 |
| 00460E6C | 12 0C A0 00 22 0C A0 00 33 0C A0 00 52 0C A0 00 | 00000000000000000000000000000000 |
| 00460E7C | 6C 0C A0 00 7D 0C A0 00 8E 0C A0 00 9F 0C A0 00 | 00000000000000000000000000000000 |
| 00460E8C | F7 A8 B1 76 00 00 00 00 C8 74 F8 72 73 66 F9 72 | 00000000000000000000000000000000 |
| 00460E9C | 87 72 F8 72 43 80 F8 72 67 37 F9 72 FB 41 F9 72 | 00000000000000000000000000000000 |
| 00460EAC | 67 83 F8 72 90 53 F8 72 00 00 00 00 CE 00 37 76 | 00000000000000000000000000000000 |
| 00460EBC | 7C 86 37 76 B0 86 37 76 33 25 36 76 1E 31 36 76 | 00000000000000000000000000000000 |
| 00460ECC | D8 7C 37 76 89 C2 37 76 CD 46 38 76 CE EE 36 76 | 00000000000000000000000000000000 |
| 00460EDC | 00 00 00 00 48 F0 4C 77 9C CB 4D 77 CC 42 4F 77 | 00000000000000000000000000000000 |
| 00460EEC | 2C 00 4C 77 DA F6 4C 77 73 33 50 77 10 64 4D 77 | 00000000000000000000000000000000 |

湛蓝色标注出来的这部分形式为 Axxxxxx 的地址是被重定向过的,被重定向到了壳的代码中,我们继续往上定位 IAT 的起始位置。

| Address | Hex dump | ASCII |
|----------|---|-------|
| 0046099C | C1 00 9F 00 D2 00 9F 00 E4 00 9F 00 F4 00 9F 00 | ±.f. |
| 004609AC | 05 01 9F 00 24 01 9F 00 3E 01 9F 00 4F 01 9F 00 | ±0f. |
| 004609BC | 60 01 9F 00 71 01 9F 00 7F 01 9F 00 8E 01 9F 00 | '0f. |
| 004609CC | 9D 01 9F 00 BD 01 9F 00 CB 01 9F 00 EE 01 9F 00 | 00f. |
| 004609DC | FF 01 9F 00 10 02 9F 00 22 02 9F 00 32 02 9F 00 | 0f. |
| 004609EC | 43 02 9F 00 62 02 9F 00 7C 02 9F 00 8D 02 9F 00 | C0f. |
| 004609FC | 9E 02 9F 00 AF 02 9F 00 BD 02 9F 00 CC 02 9F 00 | ×0f. |
| 00460A0C | DB 02 9F 00 FB 02 9F 00 09 03 9F 00 2C 03 9F 00 | ■0f. |
| 00460A1C | 3D 03 9F 00 4E 03 9F 00 60 03 9F 00 70 03 9F 00 | ÷0f. |
| 00460A2C | 81 03 9F 00 A0 03 9F 00 BA 03 9F 00 CB 03 9F 00 | ü0f. |
| 00460A3C | DC 03 9F 00 ED 03 9F 00 FB 03 9F 00 0A 04 9F 00 | ■0f. |
| 00460A4C | 19 04 9F 00 39 04 9F 00 47 04 9F 00 6A 04 9F 00 | ↓0f. |
| 00460A5C | 7B 04 9F 00 8C 04 9F 00 9E 04 9F 00 AE 04 9F 00 | (0f. |
| 00460A6C | BF 04 9F 00 DE 04 9F 00 F8 04 9F 00 09 05 9F 00 | γ0f. |
| 00460A7C | 1A 05 9F 00 2B 05 9F 00 39 05 9F 00 48 05 9F 00 | +0f. |
| 00460A8C | 57 05 9F 00 77 05 9F 00 85 05 9F 00 A8 05 9F 00 | W0f. |
| 00460A9C | B9 05 9F 00 CA 05 9F 00 DC 05 9F 00 EC 05 9F 00 | 0f. |
| 00460AAC | FD 05 9F 00 1C 06 9F 00 36 06 9F 00 47 06 9F 00 | ²0f. |
| 00460ABC | 58 06 9F 00 69 06 9F 00 77 06 9F 00 86 06 9F 00 | X0f. |
| 00460ACC | 95 06 9F 00 B5 06 9F 00 C3 06 9F 00 E6 06 9F 00 | ó0f. |
| 00460ADC | F7 06 9F 00 08 07 9F 00 1A 07 9F 00 2A 07 9F 00 | ÷0f. |
| 00460AEC | 3B 07 9F 00 5A 07 9F 00 74 07 9F 00 85 07 9F 00 | ¡0f. |
| 00460AFC | 96 07 9F 00 A7 07 9F 00 B5 07 9F 00 C4 07 9F 00 | ü0f. |
| 00460B0C | D3 07 9F 00 F3 07 9F 00 01 08 9F 00 24 08 9F 00 | É0f. |
| 00460B1C | 35 08 9F 00 46 08 9F 00 58 08 9F 00 68 08 9F 00 | 50f. |
| 00460B2C | 79 08 9F 00 98 08 9F 00 B2 08 9F 00 C3 08 9F 00 | y0f. |
| 00460B3C | D4 08 9F 00 E5 08 9F 00 F3 08 9F 00 02 09 9F 00 | É0f. |
| 00460B4C | 11 09 9F 00 31 09 9F 00 3F 09 9F 00 62 09 9F 00 | ◄0f. |
| 00460B5C | 73 09 9F 00 84 09 9F 00 96 09 9F 00 A6 09 9F 00 | s0f. |
| 00460B6C | B7 09 9F 00 D6 09 9F 00 F0 09 9F 00 01 0A 9F 00 | A0f. |
| 00460B7C | 12 0A 9F 00 23 0A 9F 00 31 0A 9F 00 40 0A 9F 00 | \$.f. |
| 00460B8C | 4F 0A 9F 00 6F 0A 9F 00 7D 0A 9F 00 A0 0A 9F 00 | O0f. |
| 00460B9C | B1 0A 9F 00 C2 0A 9F 00 D4 0A 9F 00 C0 4B 0F 77 | Σ.f. |
| 00460BAC | 3B 4C 0F 77 94 A5 11 77 59 4B 0F 77 82 4E 0F 77 | ¡L%u |
| 00460BBC | 98 D4 11 77 9B 50 0F 77 4F 50 0F 77 10 50 0F 77 | üE%u |
| 00460BCC | 3F 50 0F 77 D9 66 0F 77 50 4B 0F 77 55 4C 0F 77 | ?P%u |
| 00460BDC | C2 4B 0F 77 95 D2 11 77 80 5D 15 77 00 00 00 00 | TK%u |
| 00460BEC | 00 00 A7 00 11 00 A7 00 22 00 A7 00 33 00 A7 00 | ..0. |
| 00460BFC | 00 00 A0 00 11 00 A0 00 22 00 A0 00 33 00 A0 00 | ..0. |

这部分湛蓝色标注出来的,有几项是属于系统 DLL,其他项形式为 9Fxxxx,指向了壳创建的内存单元中。

| Address | Disassembly | Comment |
|----------|-------------------------------|------------------------|
| 0042DB98 | CALL DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 0042DBBF | CALL DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 0043070A | CALL DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 00430731 | CALL DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 0043170F | CALL DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 00431752 | CALL DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 00433301 | CALL DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 004342EE | CALL DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 00434336 | CALL DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 00434391 | CALL DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 0043440D | CALL DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 00434B04 | CALL DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 00434B33 | CALL DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 004355E0 | MOV EBP,DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 0043C0A4 | CALL DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 0043DB25 | CALL DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 0044668D | MOV EDI,DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |
| 004466D3 | MOV EDI,DWORD PTR DS:[460AC0] | DS:[00460AC0]=009F0669 |

我们来看看 460AC0 这一项参考引用,的确是重定向到壳刚刚创建的内存块中,我们继续往上看。

| Address | Hex dump | ASCII |
|----------|---|-------------------------|
| 004607BC | 62 28 06 00 80 28 06 00 70 28 06 00 00 00 00 00 | b(.,.(.p(.,.... |
| 004607CC | 0C 2B 06 00 FA 2A 06 00 E8 2A 06 00 1E 2B 06 00 | .+.,**.,p*.,.+. |
| 004607DC | 08 2A 06 00 C6 2A 06 00 AE 2A 06 00 92 2A 06 00 | i*.,*(.,*(.,*(., |
| 004607EC | 72 2A 06 00 BC 2B 06 00 A8 2B 06 00 92 2B 06 00 | r*.,.+.,.(.,+., |
| 004607FC | 78 2B 06 00 60 2B 06 00 4C 2B 06 00 2E 2B 06 00 | x+.,'+.,L+.,+. |
| 0046080C | 00 00 00 00 00 00 00 00 00 00 00 00 F0 6B DA 77 |Q.....-k rw |
| 0046081C | 1B 76 DA 77 F4 EA DA 77 E7 EB DA 77 83 78 DA 77 | +v rw 70 rw 60 rw 50 rw |
| 0046082C | 00 00 00 00 DD 15 C5 58 2E BD C3 58 00 00 00 00 |!\$+X.cX.... |
| 0046083C | 00 00 A1 00 11 00 A1 00 22 00 A1 00 33 00 A1 00 | ...l.4.l.".3.l. |
| 0046084C | 41 00 A1 00 50 00 A1 00 5F 00 A1 00 7F 00 A1 00 | A.l.P.l.".l.3.l. |
| 0046085C | 8D 00 A1 00 B0 00 A1 00 C1 00 A1 00 D2 00 A1 00 | l.l.3.l.l.l.l.l. |
| 0046086C | E4 00 A1 00 F4 00 A1 00 05 01 A1 00 24 01 A1 00 | s.l.l.l.l.l.l.l. |
| 0046087C | 3E 01 A1 00 4F 01 A1 00 60 01 A1 00 71 01 A1 00 | >0i.00i.'0i.q0i. |
| 0046088C | 7F 01 A1 00 8E 01 A1 00 9D 01 A1 00 BD 01 A1 00 | 00i.00i.00i.c0i. |
| 0046089C | CB 01 A1 00 EE 01 A1 00 FF 01 A1 00 10 02 A1 00 | 70i.'0i.0i.00i. |
| 004608AC | 22 02 A1 00 32 02 A1 00 43 02 A1 00 62 02 A1 00 | "0i.20i.C0i.b0i. |
| 004608BC | 7C 02 A1 00 8D 02 A1 00 9E 02 A1 00 AF 02 A1 00 | '0i.l0i.x0i.>0i. |
| 004608CC | BD 02 A1 00 CC 02 A1 00 DB 02 A1 00 FB 02 A1 00 | c0i.l0i.00i.'0i. |
| 004608DC | 09 03 A1 00 2C 03 A1 00 3D 03 A1 00 4E 03 A1 00 | .0i.'0i.=0i.N0i. |
| 004608EC | 60 03 A1 00 70 03 A1 00 81 03 A1 00 A0 03 A1 00 | '0i.p0i.u0i.00i. |
| 004608FC | BA 03 A1 00 CB 03 A1 00 DC 03 A1 00 ED 03 A1 00 | 0i.70i.m0i.Y0i. |
| 0046090C | FB 03 A1 00 0A 04 A1 00 19 04 A1 00 39 04 A1 00 | '0i.l.0i.l.0i.90i. |
| 0046091C | 47 04 A1 00 6A 04 A1 00 7B 04 A1 00 8C 04 A1 00 | G0i.j0i.C0i.I0i. |
| 0046092C | 9E 04 A1 00 AE 04 A1 00 BF 04 A1 00 DE 04 A1 00 | x0i.<0i.70i.i0i. |
| 0046093C | F8 04 A1 00 09 05 A1 00 1A 05 A1 00 2B 05 A1 00 | 00i.'0i.+0i.+0i. |
| 0046094C | 39 05 A1 00 48 05 A1 00 57 05 A1 00 77 05 A1 00 | 90i.H0i.W0i.W0i. |
| 0046095C | 85 05 A1 00 A8 05 A1 00 B9 05 A1 00 CA 05 A1 00 | 00i.l.0i.l.0i.00i. |
| 0046096C | DC 05 A1 00 EC 05 A1 00 00 00 9F 00 11 00 9F 00 | m0i.00i...f.4.f. |
| 0046097C | 22 00 9F 00 33 00 9F 00 41 00 9F 00 50 00 9F 00 | "f.3.f.A.f.P.f. |
| 0046098C | 5F 00 9F 00 7F 00 9F 00 8D 00 9F 00 B0 00 9F 00 | ..f.0.f.l.f.3.f. |
| 0046099C | C1 00 9F 00 D2 00 9F 00 E4 00 9F 00 F4 00 9F 00 | l.f.l.f.s.f.7.f. |
| 004609AC | 05 01 9F 00 24 01 9F 00 3E 01 9F 00 4F 01 9F 00 | 00f.00f.>0f.00f. |
| 004609BC | 60 01 9F 00 71 01 9F 00 7F 01 9F 00 8E 01 9F 00 | '0f.q0f.00f.00f. |
| 004609CC | 9D 01 9F 00 BD 01 9F 00 CB 01 9F 00 EE 01 9F 00 | 00f.c0f.70f.'0f. |
| 004609DC | FF 01 9F 00 10 02 9F 00 22 02 9F 00 32 02 9F 00 | 0f.00f."0f.20f. |
| 004609EC | 43 02 9F 00 62 02 9F 00 7C 02 9F 00 8D 02 9F 00 | C0f.b0f.l0f.l0f. |
| 004609FC | 9E 02 9F 00 AF 02 9F 00 BD 02 9F 00 CC 02 9F 00 | x0f.>0f.c0f.l0f. |
| 00460A0C | DB 02 9F 00 FB 02 9F 00 09 03 9F 00 2C 03 9F 00 | 00f.'0f.'0f.'0f. |
| 00460A1C | 3D 03 9F 00 4E 03 9F 00 60 03 9F 00 7B 03 9F 00 | =0f.N0f.'0f.p0f. |
| 00460A2C | 81 03 9F 00 A0 03 9F 00 BA 03 9F 00 CB 03 9F 00 | u0f.00f. 0f.70f. |
| 00460A3C | DC 03 9F 00 ED 03 9F 00 FB 03 9F 00 0A 04 9F 00 | m0f.Y0f.'0f.00f. |
| 00460A4C | 19 04 9F 00 39 04 9F 00 47 04 9F 00 6A 04 9F 00 | 00f.90f.G0f.J0f. |
| 00460A5C | 7B 04 9F 00 9C 04 9F 00 9F 04 9F 00 0F 04 9F 00 | C0f.l0f.x0f.<0f.'0f. |

这里用橙色标注出来的项,其中形式为 A1xxxx 的地址属于壳创建的另一个内存块。

| | | | | | | | | |
|----------|----------|--|--|--|------|----|----|--|
| 00540000 | 00103000 | | | | Map | R | R | |
| 00650000 | 00106000 | | | | Map | R | R | |
| 00970000 | 00001000 | | | | Priv | RW | RW | |
| 009F0000 | 00002000 | | | | Priv | RW | RW | |
| 00A00000 | 00002000 | | | | Priv | RW | RW | |
| 00A10000 | 00001000 | | | | Priv | RW | RW | |
| 00A20000 | 00004000 | | | | Priv | RW | RW | |
| 00A30000 | 00003000 | | | | Map | R | R | |
| 00A40000 | 00004000 | | | | Priv | RW | RW | |
| 00A50000 | 00003000 | | | | Priv | RW | RW | |
| 00A60000 | 00002000 | | | | Map | R | R | |

我们看看 460894 这一项的参考引用。

| Address | Disassembly | Comment |
|----------|----------------------------|------------------------|
| 00404F8E | CALL DWORD PTR DS:[460894] | DS:[00460894]=00A1019D |

我们继续往下,我们看到这里:

| | | |
|----------|---|-------------------------|
| 004607FC | 78 2B 06 00 60 2B 06 00 4C 2B 06 00 2E 2B 06 00 | x+.,'+.,L+.,+. |
| 0046080C | 00 00 00 00 00 00 00 00 00 00 00 00 F0 6B DA 77 |Q.....-k rw |
| 0046081C | 1B 76 DA 77 F4 EA DA 77 E7 EB DA 77 83 78 DA 77 | +v rw 70 rw 60 rw 50 rw |
| 0046082C | 00 00 00 00 DD 15 C5 58 2E BD C3 58 00 00 00 00 |!\$+X.cX.... |
| 0046083C | 00 00 A1 00 11 00 A1 00 22 00 A1 00 33 00 A1 00 | ...l.4.l.".3.l. |
| 0046084C | 41 00 A1 00 50 00 A1 00 5F 00 A1 00 7F 00 A1 00 | A.l.P.l.".l.3.l. |
| 0046085C | 8D 00 A1 00 B0 00 A1 00 C1 00 A1 00 D2 00 A1 00 | l.l.3.l.l.l.l.l. |

这几项用橙色标注出来的是属于系统 DLL 的。

| Address | Hex dump | ASCII |
|----------|---|------------------------------|
| 004607F8 | 92 2B 06 00 78 2B 06 00 60 2B 06 00 4C 2B 06 00 | AE+.,'+.,L+.,+. |
| 00460808 | 2E 2B 06 00 00 00 00 00 00 00 00 00 80 00 00 00 | .+.,*(.,*(.,*(.,*(., |
| 00460818 | FA 6B DA 77 1B 76 DA 77 F4 EA DA 77 E7 EB DA 77 | -k rw+v rw 70 rw 60 rw 50 rw |
| 00460828 | 83 78 DA 77 00 00 00 00 DD 15 C5 58 2E BD C3 58 | 00 rw....!\$+X.cX.... |
| 00460838 | 00 00 00 00 00 00 A1 00 11 00 A1 00 22 00 A1 00 | ...l.4.l.".3.l. |
| 00460848 | 33 00 A1 00 41 00 A1 00 50 00 A1 00 5F 00 A1 00 | s.l.A.l.P.l.".l.3.l. |
| 00460858 | 7F 00 A1 00 8D 00 A1 00 B0 00 A1 00 C1 00 A1 00 | 0.l.l.l.3.l.l.l.l. |
| 00460868 | D2 00 A1 00 E4 00 A1 00 F4 00 A1 00 05 01 A1 00 | E.l.l.s.l.l.l.l.l. |
| 00460878 | 24 01 A1 00 3E 01 A1 00 4F 01 A1 00 60 01 A1 00 | \$0i.>0i.00i.'0i. |
| 00460888 | 71 01 A1 00 7F 01 A1 00 8E 01 A1 00 9F 01 A1 00 | q0i.00i.00i.'0i. |
| 00460898 | BD 01 A1 00 CB 01 A1 00 EE 01 A1 00 FF 01 A1 00 | c0i.70i.00i.00i. |
| 004608A8 | 10 02 A1 00 22 02 A1 00 32 02 A1 00 43 02 A1 00 | 00i.'0i.20i.C0i. |
| 004608B8 | 62 02 A1 00 7C 02 A1 00 8D 02 A1 00 9E 02 A1 00 | b0i.l0i.l0i.x0i. |
| 004608C8 | AF 02 A1 00 BD 02 A1 00 CC 02 A1 00 DB 02 A1 00 | >0i.c0i.l0i.00i. |
| 004608D8 | FB 02 A1 00 09 03 A1 00 2C 03 A1 00 3D 03 A1 00 | '0i.'0i.'0i.'0i. |
| 004608E8 | 4E 03 A1 00 60 03 A1 00 7B 03 A1 00 8C 03 A1 00 | N0i.'0i.N0i.u0i. |
| 004608F8 | A0 03 A1 00 BA 03 A1 00 CB 03 A1 00 DC 03 A1 00 | 00i. 0i.70i.00i. |
| 00460908 | ED 03 A1 00 FB 03 A1 00 0A 04 A1 00 19 04 A1 00 | Y0f.'0f.'0f.'0f. |
| 00460918 | 39 04 A1 00 47 04 A1 00 6A 04 A1 00 7B 04 A1 00 | 90i.G0i.J0i.C0i. |
| 00460928 | 8C 04 A1 00 9F 04 A1 00 9F 04 A1 00 0F 04 A1 00 | C0f.x0i.<0i.'0f. |

再往下就是零了,左边一项是 80000000,明显不属于任何一个内存单元。

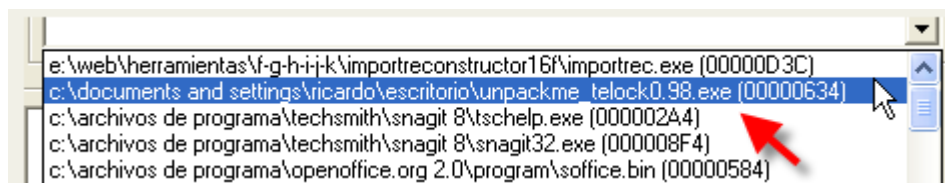
所以 IAT 的起始地址为 460818,长度为 710,OEP 为 4271B0。

OEP = 271B0(RVA)

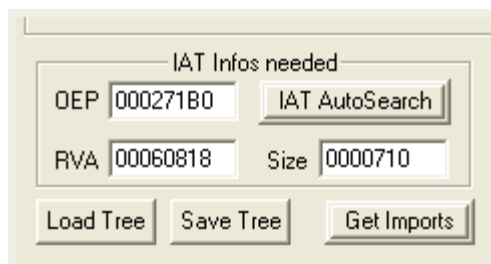
IAT 起始地址 = 60818(RVA)

IAT 长度 = 710

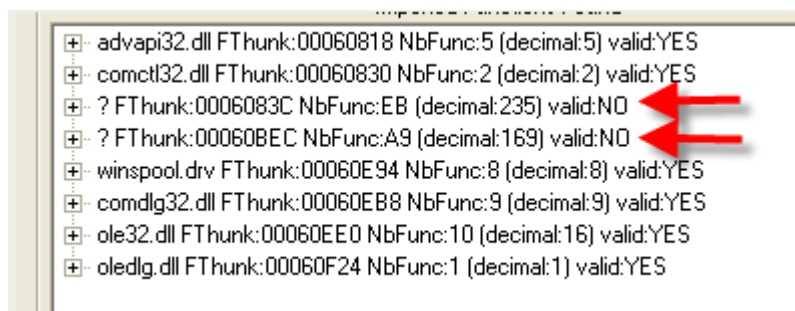
我们打开 IMP REC,定位到 telock0.98 所在的进程。



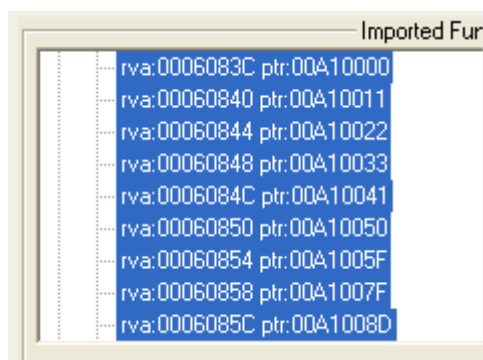
将 OEP,IAT 起始地址,长度的值都填上。



单击 Get Imports。



我们可以看到 IMP REC 检测到了重定向过的项,但是提示无效,我们单击右边的 Show Invalids(显示无效的项)。



关于这些重定向的 IAT 项如何修复我们在下一章节继续讨论。