[原创]带狗用x64dbg脱SafeNet Sentinel LDK加密的64位VC程序

csjwaman 5 24

专家

2020-2-19 11:28 8865

Take the dog to use x64dbg to remove the 64-bit VC program encrypted by SafeNet Sentinel LDK

Run the program without a dog, prompt:

```
---------------------------
Sentinel LDK Protection System
---------------------------
Sentinel key not found (H0007)
---------------------------
determine
---------------------------
```

ExeinfoPe.exe checks the shell and shows: 64 bit executable -> Unknown Packer-Protector, 4 sections / Self Write code? Unknown Protection

Detect It Easy 2.02 Lite checks the shell and shows:
PE+(64): protector: SafeNet Sentinel LDK(-)[-]
PE+(64): linker: Microsoft Linker(12.0)[EXE64,admin]

Plug in the dog below and start taking the dog out of the shell.

1. Handle program relocation

Since the program code base address will be relocated, it is troublesome to deal with the relocation problem after shelling.
So you have to modify the program DllCharacteristices first, change its value from the original 0x8160 to 0x8120, that is, cancel the DLL attribute. This will not relocate.
2. Document verification

After modifying the file, the shell will detect the file change.
Load the program with x64dbg, run it directly, and pop up

```
---------------------------
Sentinel LDK Protection System
---------------------------
Internal error 0x7102 occurred!
---------------------------
determine
---------------------------
```

Pause the program and view the call stack:

```
000000000012F798 0000000077807214 00000000777FBF5A 90 user32.00000000777FBF5A system m
```

```
000000000012F828 00000000778074A5 0000000077807214 60 user32.0000000077807214 system module
000000000012F888 00000000778527F0 00000000778074A5 130 user32.00000000778074A5 system module
000000000012F9B8 0000000077851AE5 00000000778527F0 1C0 user32.00000000778527F0 system module
000000000012FB78 000000007785133B 0000000077851AE5 D0 user32.0000000077851AE5 system module
000000000012FC48 0000000077851232 000000007785133B 40 user32.000000007785133B System module
000000000012FC88 000000014085AD21 0000000077851232 8 user32.0000000077851232 User module
000000000012FC90 00000000778511E3 000000014085AD21 40 studype_2.000000014085AD21 system module //Go to this address to see
000000000012FCD0 0000000000000002 00000000778511E3 8 user32.00000000778511E3 User module
000000000012FCD8 000000014085CF8F 0000000000000002 8 0000000000000002 User module
000000000012FCE0 0000000000000000 000000014085CF8F studype_2.000000014085CF8F user module
```

```
000000014085AD15 | 4C:8BC6 | MOV R8,RSI |
000000014085AD18 | 48:8BD5 | MOV RDX,RBP |
000000014085AD1B | 48:8BCB | MOV RCX,RBX |
000000014085AD1E | 41:FFD3 | CALL R11 |
000000014085AD21 | EB 02 | JMP 0x14085AD25 | Here is the address displayed in the stack, look up for the entrance
000000014085AD23 | 33C0 | XOR EAX,EAX |
000000014085AD25 | 48:83C4 28 | ADD RSP,0x28 |
000000014085AD29 | 5F | POP RDI |
000000014085AD2A | 5E | POP RSI |
000000014085AD2B | 5D | POP RBP |
000000014085AD2C | 5B | POP RBX |
000000014085AD2D | C3 | RET |//Back
```

```
000000014085CF30 | 44:0FB69C24 F801000 | MOVZX R11D,BYTE PTR SS:[RSP+0x1F8] |
000000014085CF39 | 49:8BC5 | MOV RAX,R13 |
000000014085CF3C | 66:B9 0300 | MOV CX,0x3 |
000000014085CF40 | 49:F7E3 | MUL R11 |
000000014085CF43 | 48:C1EA 02 | SHR RDX,0x2 |
000000014085CF47 | 48:8D0492 | LEA RAX,QWORD PTR DS:[RDX+RDX*4] |
000000014085CF4B | 4C:2BD8 | SUB R11,RAX |
000000014085CF4E | 41:0FB6C3 | MOVZX EAX,R11B |
000000014085CF52 | 44:889C24 F8010000 | MOV BYTE PTR SS:[RSP+0x1F8],R11B |
000000014085CF5A | 41:FF14C6 | CALL QWORD PTR DS:[R14+RAX*8] |
000000014085CF5E | 392D C8A90600 | CMP DWORD PTR DS:[0x1408C792C],EBP |
000000014085CF64 | 74 2E | JE 0x14085CF94 |
000000014085CF66 | 48:8BCE | MOV RCX,RSI |
000000014085CF69 | E8 3AC8FFFF | CALL 0x1408597A8 | Enter this CALL
000000014085CF6E | 3BC5 | CMP EAX,EBP |
000000014085CF70 | 8BF8 | MOV EDI,EAX |
000000014085CF72 | 74 20 | JE 0x14085CF94 | Here you can skip the error message
000000014085CF74 | 4C:8D05 D5660500 | LEA R8,QWORD PTR DS:[0x1408B3650] | 00000001408B3650:L"Sentinel LDK Protection System"
000000014085CF7B | 48:8D15 4E670500 | LEA RDX,QWORD PTR DS:[0x1408B36D0] | 00000001408B36D0:L"Internal error 0x7102 occurred!"
```

```
000000014085CF8A | E8 11DDFFFF | CALL 0x14085ACA0 |
000000014085CF8F | E9 EC030000 | JMP 0x14085D380 | Return to here, look up and find an error message
000000014085CF94 | 0FBA25 6C900600 0C | BT DWORD PTR DS:[0x1408C6008],0xC |
000000014085CF9C | 72 71 | JB 0x14085D00F |

Enter CALL 0x1408597A8:
00000001408597A8 | 48:53 | PUSH RBX |
00000001408597AA | 55 | PUSH RBP |
00000001408597AB | 56 | PUSH RSI |
00000001408597AC | 57 | PUSH RDI |
00000001408597AD | 48:8BC4 | MOV RAX,RSP |
00000001408597B0 | 48:81EC 58010000 | SUB RSP,0x158 |
00000001408597B7 | 48:8BD9 | MOV RBX,RCX |
00000001408597BA | 33FF | XOR EDI,EDI |
00000001408597BC | 48:8D4C24 41 | LEA RCX,QWORD PTR SS:[RSP+0x41] |
00000001408597C1 | 33D2 | XOR EDX,EDX |
00000001408597C3 | 41:B8 03010000 | MOV R8D,0x103 |
00000001408597C9 | 33ED | XOR EBP,EBP |
00000001408597CB | 8978 30 | MOV DWORD PTR DS:[RAX+0x30],EDI |
00000001408597CE | 40:887C24 40 | MOV BYTE PTR SS:[RSP+0x40],DIL |
00000001408597D3 | E8 580F0500 | CALL 0x1408AA730 |
00000001408597D8 | BE 04010000 | MOV ESI,0x104 |
00000001408597DD | 48:8D5424 40 | LEA RDX,QWORD PTR SS:[RSP+0x40] |
00000001408597E2 | 48:8BCB | MOV RCX,RBX |
00000001408597E5 | 44:8BC6 | MOV R8D,ESI |
00000001408597E8 | E8 47110500 | CALL <JMP.&GetModuleFileNameA> |
00000001408597ED | 3BC6 | CMP EAX,ESI |
00000001408597EF | 0F84 2F010000 | JE 0x140859924 |
00000001408597F5 | 85C0 | TEST EAX,EAX |
00000001408597F7 | 0F84 27010000 | JE 0x140859924 |
00000001408597FD | 48:897C24 30 | MOV QWORD PTR SS:[RSP+0x30],RDI | [rsp+30]: "0 return"
0000000140859802 | 8D5F 01 | LEA EBX,QWORD PTR DS:[RDI+0x1] |
0000000140859805 | 48:8D4C24 40 | LEA RCX,QWORD PTR SS:[RSP+0x40] |
000000014085980A | 45:33C9 | XOR R9D,R9D |
000000014085980D | BA 00000080 | MOV EDX,0x80000000 |
0000000140859812 | 44:8BC3 | MOV R8D,EBX |
0000000140859815 | C74424 28 80000000 | MOV DWORD PTR SS:[RSP+0x28],0x80 |
000000014085981D | C74424 20 03000000 | MOV DWORD PTR SS:[RSP+0x20],0x3 |
0000000140859825 | E8 04110500 | CALL <JMP.&CreateFileA> | Open program file (file on disk, not in memory)
000000014085982A | 48:83F8 FF | CMP RAX,0xFFFFFFFFFFFFFFFF |
000000014085982E | 48:8BF8 | MOV RDI,RAX |
0000000140859831 | 0F84 F2000000 | JE 0x140859929 |
0000000140859837 | 33D2 | XOR EDX,EDX |
0000000140859839 | 48:8BC8 | MOV RCX,RAX |
000000014085983C | E8 E7100500 | CALL <JMP.&GetFileSize> | Get the file size on the disk
0000000140859841 | 8BF0 | MOV ESI,EAX |
0000000140859843 | B8 FFFFFFFF | MOV EAX,0xFFFFFFFF |
0000000140859848 | 48:3BF0 | CMP RSI,RAX |
000000014085984B | 0F84 D8000000 | JE 0x140859929 |
0000000140859851 | 48:8BCE | MOV RCX,RSI |
0000000140859854 | E8 57A30000 | CALL 0x140863BB0 |
0000000140859859 | 48:85C0 | TEST RAX,RAX |
000000014085985C | 48:8BE8 | MOV RBP,RAX |
000000014085985F | 75 08 | JNE 0x140859869 |
0000000140859869 | 4C:8D8C24 88010000 | LEA R9,QWORD PTR SS:[RSP+0x188] |
0000000140859871 | 44:8BC6 | MOV R8D,ESI |
0000000140859874 | 48:8BD0 | MOV RDX,RAX |
0000000140859877 | 48:8BCF | MOV RCX, RDI |
000000014085987A | 48:C74424 20 000000 | MOV QWORD PTR SS:[RSP+0x20],0x0 |
0000000140859883 | E8 9A100500 | CALL <JMP.&ReadFile> | Read the file on the disk
0000000140859888 | 85C0 | TEST EAX,EAX |
000000014085988A | 0F84 94000000 | JE 0x140859924 |
0000000140859890 | 8B8424 88010000 | MOV EAX,DWORD PTR SS:[RSP+0x188] |
0000000140859897 | 48:3BC6 | CMP RAX,RSI |
000000014085989A | 0F85 84000000 | JNE 0x140859924 |
00000001408598A0 | 8B05 66C70600 | MOV EAX,DWORD PTR DS:[0x1408C600C] |
00000001408598A6 | 33D2 | XOR EDX,EDX |
00000001408598A8 | 33C9 | XOR ECX,ECX |
00000001408598AA | 898424 94010000 | MOV DWORD PTR SS:[RSP+0x194],EAX |
00000001408598B1 | 898424 90010000 | MOV DWORD PTR SS:[RSP+0x190],EAX |
00000001408598B8 | 8B05 6EE00600 | MOV EAX,DWORD PTR DS:[0x1408C792C] |
00000001408598BE | 85C0 | TEST EAX,EAX |
00000001408598C0 | 7E 42 | JLE 0x140859904 |
00000001408598C2 | 4C:8D05 63D80600 | LEA R8,QWORD PTR DS:[0x1408C712C] |
00000001408598C9 | 48:83F9 08 | CMP RCX,0x8 |
00000001408598CD | 75 19 | JNE 0x1408598E8 |
00000001408598CF | 33C9 | XOR ECX,ECX |
00000001408598D1 | 41:8A4408 F8 | MOV AL,BYTE PTR DS:[R8+RCX-0x8] |
00000001408598D6 | 00840C 90010000 | ADD BYTE PTR SS:[RSP+RCX+0x190],AL |
00000001408598DD | 48:03CB | ADD RCX,RBX |
00000001408598E0 | 48:83F9 08 | CMP RCX,0x8 |
00000001408598E4 | 7C EB | JL 0x1408598D1 |
00000001408598E6 | 33C9 | XOR ECX,ECX |
00000001408598E8 | 8A840C 90010000 | MOV AL,BYTE PTR SS:[RSP+RCX+0x190] |
00000001408598EF | 03D3 | ADD EDX,EBX |
00000001408598F1 | 48:03CB | ADD RCX,RBX |
00000001408598F4 | 41:3000 | XOR BYTE PTR DS:[R8],AL |
00000001408598F7 | 8B05 2FE00600 | MOV EAX,DWORD PTR DS:[0x1408C792C] |
00000001408598FD | 4C:03C3 | ADD R8,RBX |
0000000140859900 | 3BD0 | CMP EDX,EAX |
0000000140859902 | 7C C5 | JL 0x1408598C9 |
0000000140859904 | 4C:8D05 21D80600 | LEA R8,QWORD PTR DS:[0x1408C712C] |
000000014085990B | 44:8BC8 | MOV R9D,EAX |
000000014085990E | 48:8BD6 | MOV RDX,RSI |
0000000140859911 | 48:8BCD | MOV RCX,RBP |
0000000140859914 | E8 47BF0000 | CALL 0x140865860 |
0000000140859919 | F7D8 | NEG EAX |
000000014085991B | 1BC9 | SBB ECX,ECX |
000000014085991D | 83E1 04 | AND ECX,0x4 |
0000000140859920 | 8BD9 | MOV EBX,ECX |
0000000140859922 | EB 05 | JMP 0x140859929 |
0000000140859924 | BB 03000000 | MOV EBX,0x3 |
0000000140859929 | 44:8B05 FCDF0600 | MOV R8D,DWORD PTR DS:[0x1408C792C] |
0000000140859930 | 48:8D0D F5D70600 | LEA RCX,QWORD PTR DS:[0x1408C712C] |
0000000140859937 | 33D2 | XOR EDX,EDX |
0000000140859939 | E8 52A30000 | CALL 0x140863C90 |
000000014085993E | 48:85ED | TEST RBP,RBP |
0000000140859941 | C705 E1DF0600 00000 | MOV DWORD PTR DS:[0x1408C792C],0x0 |
000000014085994B | 74 08 | JE 0x140859955 |
000000014085994D | 48:8BCD | MOV RCX,RBP |
0000000140859958 | 74 0E | JE 0x140859968 |
000000014085995A | 48:83FF FF | CMP RDI,0xFFFFFFFFFFFFFFFF |
000000014085995E | 74 08 | JE 0x140859968 |
0000000140859960 | 48:8BCF | MOV RCX,RDI |
0000000140859963 | E8 B40F0500 | CALL <JMP.&CloseHandle> |
0000000140859968 | 8BC3 | MOV EAX,EBX |
000000014085996A | 48:81C4 58010000 | ADD RSP,0x158 |
0000000140859971 | 5F | POP RDI |
0000000140859972 | 5E | POP RSI |
0000000140859973 | 5D | POP RBP |
0000000140859974 | 5B | POP RBX |
0000000140859975 | C3 | RET |
```

It can be found that this is a verification CALL, and it can be verified by returning 0 directly.

3. Skip the IAT encryption and search program OEP

15 API addresses of this program are encrypted. Encryption needs to be skipped.

VirtualProtect conditional breakpoint: rcx>StudyPE:0 && rcx<StudyPE:0 + 0xe4e000

StudyPE is the name of the program being debugged, and 0xe4e000 is the image size of the entire program

IAT table starting address: studype:0 + 0x379000

Break 2 times and return:

```
00000001405D7CF1 | 49:8BD4 | MOV RDX,R12 |
00000001405D7CF4 | 48:8BCD | MOV RCX,RBP |
00000001405D7CF7 | 8B442F 4C | MOV EAX,DWORD PTR DS:[RDI+RBP+0x4C] |
00000001405D7CFB | 48:8BF3 | MOV RSI,RBX |
00000001405D7CFE | 8905 08830700 | MOV DWORD PTR DS:[0x14065000C],EAX |
00000001405D7D04 | 44:8BB5 90000000 | MOV R14D,DWORD PTR SS:[RBP+0x90] |
00000001405D7D0B | 4D:03F7 | ADD R14,R15 |
00000001405D7D0E | E8 FD2B0500 | CALL <JMP.&VirtualProtect> | First call
00000001405D7D13 | 41:3BC5 | CMP EAX,R13D |
00000001405D7D16 | 74 2A | JE 0x1405D7D42 |
00000001405D7D18 | C7442F 3C 20000060 | MOV DWORD PTR DS:[RDI+RBP+0x3C],0x60000020 |
00000001405D7D20 | 4C:89AD 90000000 | MOV QWORD PTR SS:[RBP+0x90],R13 |
00000001405D7D27 | 44:8B8424 A0010000 | MOV R8D,DWORD PTR SS:[RSP+0x1A0] |
00000001405D7D2F | 4C:8D8C24 A0010000 | LEA R9,QWORD PTR SS:[RSP+0x1A0] |
00000001405D7D37 | 49:8BD4 | MOV RDX,R12 |
00000001405D7D3A | 48:8BCD | MOV RCX,RBP |
00000001405D7D3D | E8 CE2B0500 | CALL <JMP.&VirtualProtect> | Second call
00000001405D7D42 | 833B FF | CMP DWORD PTR DS:[RBX],0xFFFFFFFF | Back here
00000001405D7D45 | 48:8D3D 34590000 | LEA RDI,QWORD PTR DS:[0x1405DD680] |
00000001405D7D4C | 0F84 B8050000 | JE 0x1405D830A |
......
000000013FD37EF4 | 44:8B23 | MOV R12D,DWORD PTR DS:[RBX] |
000000013FD37EF7 | 48:8B8C24 B8010000 | MOV RCX,QWORD PTR SS:[RSP+0x1B8] |
000000013FD37EFF | 4C:8D05 FE290500 | LEA R8,QWORD PTR DS:[<JMP.&GetModuleHandleA>] |
000000013FD37F06 | 41:8BD4 | MOV EDX,R12D |
000000013FD37F09 | 41:81E1 00020000 | AND R9D,0x200 |
000000013FD37F10 | E8 0B0B0000 | CALL 0x13FD38A20 |
000000013FD37F15 | 48:8B8C24 B8010000 | MOV RCX,QWORD PTR SS:[RSP+0x1B8] |
000000013FD37F1D | 48:3B0D 14810700 | CMP RCX,QWORD PTR DS:[0x13FDB0038] |[0x13FDB0038]
```

```
000000013FD37F27 | 75 24 | JNE 0x13FD37F4D | If it is kernel32.dll JMP
000000013FD37F29 | 41:81FC 7ED8EC73 | CMP R12D,0x73ECD87E | Then compare whether it is the specified function to be encrypted
000000013FD37F30 | 75 09 | JNE 0x13FD37F3B |
000000013FD37F32 | 48:8D3D 47570000 | LEA RDI,QWORD PTR DS:[0x13FD3D680] |
000000013FD37F39 | EB 12 | JMP 0x13FD37F4D |
000000013FD37F3B | 48:8D05 82570000 | LEA RAX,QWORD PTR DS:[0x13FD3D6C4] |
000000013FD37F42 | 41:81FC 83B9BA78 | CMP R12D,0x78BAB983 | The specified function to be encrypted
000000013FD37F49 | 48:0F44F8 | CMOVE RDI,RAX |
000000013FD37F4D | F643 04 30 | TEST BYTE PTR DS:[RBX+0x4],0x30 |
000000013FD37F51 | 74 19 | JE 0x13FD37F6C |
000000013FD37F53 | 48:8BCF | MOV RCX,RDI |
000000013FD37F56 | E8 BD1B0000 | CALL 0x13FD39B18 |
000000013FD37F5B | 84C0 | TEST AL,AL |
000000013FD37F5D | 74 05 | JE 0x13FD37F64 |
000000013FD37F5F | E8 003D0000 | CALL 0x13FD3BC64 |
000000013FD37F64 | 48:8B8C24 B8010000 | MOV RCX,QWORD PTR SS:[RSP+0x1B8] |
000000013FD37F6C | C703 00000000 | MOV DWORD PTR DS:[RBX],0x0 |
000000013FD37F72 | 8B43 04 | MOV EAX,DWORD PTR DS:[RBX+0x4] |
000000013FD37F75 | 48:83C3 09 | ADD RBX,0x9 |
000000013FD37F79 | A8 10 | TEST AL,0x10 |
000000013FD37F7B | 898424 A0010000 | MOV DWORD PTR SS:[RSP+0x1A0],EAX |
000000013FD37F82 | 74 31 | JE 0x13FD37FB5 |
000000013FD37F84 | C645 00 E8 | MOV BYTE PTR SS:[RBP],0xE8 |
000000013FD37F88 | 48:FFC5 | INC RBP |
000000013FD37F8B | F605 76E00600 80 | TEST BYTE PTR DS:[0x13FDA6008],0x80 |
000000013FD37F92 | 48:8D05 2F0A0000 | LEA RAX,QWORD PTR DS:[0x13FD389C8] |
000000013FD37F99 | 75 07 | JNE 0x13FD37FA2 |
000000013FD37F9B | 48:8D05 1C070000 | LEA RAX,QWORD PTR DS:[0x13FD386BE] |
000000013FD37FA2 | 2BC5 | SUB EAX,EBP |
000000013FD37FA4 | 83E8 04 | SUB EAX,0x4 |
000000013FD37FA7 | 8945 00 | MOV DWORD PTR SS:[RBP],EAX |
000000013FD37FAA | 8B8424 A0010000 | MOV EAX,DWORD PTR SS:[RSP+0x1A0] |
000000013FD37FB1 | 48:83C5 04 | ADD RBP,0x4 |
000000013FD37FB5 | A8 20 | TEST AL,0x20 |
000000013FD37FB7 | 74 32 | JE 0x13FD37FEB |
000000013FD37FB9 | 41:C606 E8 | MOV BYTE PTR DS:[R14],0xE8 |
000000013FD37FBD | 49:FFC6 | INC R14 |
000000013FD37FC0 | F605 41E00600 80 | TEST BYTE PTR DS:[0x13FDA6008],0x80 |
000000013FD37FC7 | 48:8D05 C7060000 | LEA RAX,QWORD PTR DS:[0x13FD38695] |
000000013FD37FCE | 75 07 | JNE 0x13FD37FD7 |
000000013FD37FD0 | 48:8D05 CC080000 | LEA RAX,QWORD PTR DS:[0x13FD388A3] |
000000013FD37FD7 | 41:2BC6 | SUB EAX,R14D |
000000013FD37FDA | 83E8 04 | SUB EAX,0x4 |
000000013FD37FDD | 41:8906 | MOV DWORD PTR DS:[R14],EAX |
000000013FD37FE0 | 8B8424 A0010000 | MOV EAX,DWORD PTR SS:[RSP+0x1A0] |
000000013FD37FE7 | 49:83C6 04 | ADD R14,0x4 |
000000013FD37FEB | 48:3B0D 46800700 | CMP RCX,QWORD PTR DS:[0x13FDB0038] | Continue to compare whether it is kernel32.dll
000000013FD37FF2 | 0F85 27010000 | JNE 0x13FD3811F |JMP
000000013FD37FF8 | B9 6BE995C6 | MOV ECX,0xC695E96B |
000000013FD37FFD | 44:3BE1 | CMP R12D,ECX |
000000013FD38000 | 0F87 AD000000 | JA 0x13FD380B3 |
000000013FD38006 | 0F84 9B000000 | JE 0x13FD380A7 |
000000013FD3800C | 41:81FC 6381D90F | CMP R12D,0xFD98163 | The specified function to be encry
```

```
000000013FD38019 | 41:81FC 9EF9FB36 | CMP R12D,0x36FBF99E | The specified function to be encrypted
000000013FD38020 | 74 6D | JE 0x13FD3808F |
000000013FD38022 | 41:81FC B460326A | CMP R12D,0x6A3260B4 | The specified function to be encrypted
000000013FD38029 | 74 58 | JE 0x13FD38083 |
000000013FD3802B | 41:81FC 0E188F7B | CMP R12D,0x7B8F180E | The specified function to be encrypted
000000013FD38032 | 74 43 | JE 0x13FD38077 |
000000013FD38034 | 41:81FC 0D61778A | CMP R12D,0x8A77610D |Specified function to be encrypted
000000013FD3803B | 74 2E | JE 0x13FD3806B |
000000013FD3803D | 41:81FC AA0CB192 | CMP R12D,0x92B10CAA |Specified function to be encrypted
000000013FD38044 | 74 19 | JE 0x13FD3805F |
000000013FD38046 | 41:81FC 02E230BE | CMP R12D,0xBE30E202 | The specified function to be encrypted
000000013FD3804D | 0F85 F1000000 | JNE 0x13FD38144 |
000000013FD38053 | 48:8D3D 5A570000 | LEA RDI,QWORD PTR DS:[0x13FD3D7B4] |Remove the encrypted function address
000000013FD3805A | E9 E5000000 | JMP 0x13FD38144 |
000000013FD3805F | 48:8D3D 1E570000 | LEA RDI,QWORD PTR DS:[0x13FD3D784] |Remove the encrypted function address
000000013FD38066 | E9 D9000000 | JMP 0x13FD38144 |
000000013FD3806B | 48:8D3D BA550000 | LEA RDI,QWORD PTR DS:[0x13FD3D62C] |Remove the encrypted function address
000000013FD38072 | E9 CD000000 | JMP 0x13FD38144 |
000000013FD38077 | 48:8D3D 22550000 | LEA RDI,QWORD PTR DS:[0x13FD3D5A0] | Take out the encrypted function address
000000013FD3807E | E9 C1000000 | JMP 0x13FD38144 |
000000013FD38083 | 48:8D3D CA560000 | LEA RDI,QWORD PTR DS:[0x13FD3D754] |Remove the encrypted function address
000000013FD3808A | E9 B5000000 | JMP 0x13FD38144 |
000000013FD3808F | 48:8D3D DE540000 | LEA RDI,QWORD PTR DS:[0x13FD3D574] |Remove the encrypted function address
000000013FD38096 | E9 A9000000 | JMP 0x13FD38144 |
000000013FD3809B | 48:8D3D 72570000 | LEA RDI,QWORD PTR DS:[0x13FD3D814] |Remove the encrypted function address
000000013FD380A2 | E9 9D000000 | JMP 0x13FD38144 |
000000013FD380A7 | 48:8D3D 36570000 | LEA RDI,QWORD PTR DS:[0x13FD3D7E4] | Take out the encrypted function address
000000013FD380AE | E9 91000000 | JMP 0x13FD38144 |
000000013FD380B3 | 41:81FC B36982D8 | CMP R12D,0xD88269B3 | The specified function to be encrypted
000000013FD380BA | 74 5A | JE 0x13FD38116 |
000000013FD380BC | 41:81FC 02040EE6 | CMP R12D,0xE60E0402 | The specified function to be encrypted
000000013FD380C3 | 74 48 | JE 0x13FD3810D |
000000013FD380C5 | 41:81FC 31DD2EE6 | CMP R12D,0xE62EDD31 | The specified function to be encrypted
000000013FD380CC | 74 36 | JE 0x13FD38104 |
000000013FD380CE | 41:81FC E4CFD2E8 | CMP R12D,0xE8D2CFE4 |Specified function to be encrypted
000000013FD380D5 | 74 24 | JE 0x13FD380FB |
000000013FD380D7 | 41:81FC F543C4F1 | CMP R12D,0xF1C443F5 |Specified function to be encrypted
000000013FD380DE | 74 12 | JE 0x13FD380F2 |
000000013FD380E7 | 75 5B | JNE 0x13FD38144 |
000000013FD380E9 | 48:8D3D 10550000 | LEA RDI,QWORD PTR DS:[0x13FD3D600] |Remove the encrypted function address
000000013FD380F0 | EB 52 | JMP 0x13FD38144 |
000000013FD380F2 | 48:8D3D 4B570000 | LEA RDI,QWORD PTR DS:[0x13FD3D844] |Remove the encrypted function address
000000013FD380F9 | EB 49 | JMP 0x13FD38144 |
000000013FD380FB | 48:8D3D CE540000 | LEA RDI,QWORD PTR DS:[0x13FD3D5D0] | Take out the encrypted function address
000000013FD38102 | EB 40 | JMP 0x13FD38144 |
000000013FD38104 | 48:8D3D 1D560000 | LEA RDI,QWORD PTR DS:[0x13FD3D728] |Remove the encrypted function address
000000013FD3810B | EB 37 | JMP 0x13FD38144 |
000000013FD3810D | 48:8D3D 30540000 | LEA RDI,QWORD PTR DS:[0x13FD3D544] |Remove the encrypted function address
000000013FD38114 | EB 2E | JMP 0x13FD38144 |
000000013FD38116 | 48:8D3D 57570000 | LEA RDI,QWORD PTR DS:[0x13FD3D874] |Remove the encrypted function address
000000013FD3811D | EB 25 | JMP 0x13FD38144 |
000000013FD3811F | 48:8B5424 38 | MOV RDX,QWORD PTR SS:[RSP+0x38] |QWORD PTR SS:[RSP+0x38] is the base address of ws2_32.dll
000000013FD38124 | 48:85D2 | TEST RDX,RDX |
000000013FD38127 | 74 1B | JE 0x13FD38144 | Can't jump!
000000013FD38129 | 48:3BCA | CMP RCX,RDX |
000000013FD3812C | 75 16 | JNE 0x13FD38144 |
000000013FD3812E | 41:81FC 482CBD59 | CMP R12D,0x59BD2C48 | The specified function to be encrypted ws2_32.WSACleanup
000000013FD38135 | 74 06 | JE 0x13FD3813D |
000000013FD38137 | 41:83FC 74 | CMP R12D,0x74 |
000000013FD3813B | 75 07 | JNE 0x13FD38144 |
000000013FD3813D | 48:8D3D A0570000 | LEA RDI,QWORD PTR DS:[0x13FD3D8E4] | Take out the encrypted function address
000000013FD38144 | A8 40 | TEST AL,0x40 |
000000013FD38146 | 75 04 | JNE 0x13FD3814C |
000000013FD38148 | A8 30 | TEST AL,0x30 |
000000013FD3814A | 75 0A | JNE 0x13FD38156 |
000000013FD3814C | 48:893E | MOV QWORD PTR DS:[RSI],RDI | Fill IAT, RDI is the function address (some are encrypted addresses), RSI is the IAT address
000000013FD3814F | 8B8424 A0010000 | MOV EAX,DWORD PTR SS:[RSP+0x1A0] |
000000013FD38156 | 48:83C6 08 | ADD RSI,0x8 | Next
..... Omit part of the code.........
000000013FD3838A | F605 83DC0600 02 | TEST BYTE PTR DS:[0x13FDA6014],0x2 |
000000013FD38391 | 49:0F45FD | CMOVNE RDI,R13 |
000000013FD38395 | 48:8BC7 | MOV RAX,RDI |
000000013FD38398 | 48:81C4 58010000 | ADD RSP,0x158 |
000000013FD3839F | 41:5F | POP R15 |
000000013FD383A1 | 41:5E | POP R14 |
000000013FD383A3 | 41:5D | POP R13 |
000000013FD383A5 | 41:5C | POP R12 |
000000013FD383A7 | 5F | POP RDI |
000000013FD383A8 | 5E | POP RSI |
000000013FD383A9 | 5D | POP RBP |
000000013FD383AA | 5B | POP RBX |
000000013FD383AB | C3 | RET | Go here directly to break and return
```

Back here:

```
000000013F908E14 | 48:8D3D ADFFFFFF | LEA RDI,QWORD PTR DS:[0x13F908DC8] |
000000013F908E1B | 41:58 | POP R8 |
000000013F908E1D | B0 E9 | MOV AL,0xE9 |
000000013F908E1F | 5A | POP RDX | rdx:sub_13FDE7000
000000013F908E20 | AA | STOSB |
000000013F908E21 | 59 | POP RCX |
000000013F908E22 | B8 69000000 | MOV EAX,0x69 | 69:'i'
000000013F908E27 | 5B | POP RBX |
000000013F908E28 | AB | STOSD |
000000013F908E29 | 9D | POPFQ |
000000013F908E2A | 5E | POP RSI |
000000013F908E2B | 58 | POP RAX |
000000013F908E2C | 5F | POP RDI |
000000013F908E2D | EB 07 | JMP 0x13F908E36 |
000000013F908E2F | 48:81C4 D0000000 | ADD RSP,0xD0 |
000000013F908E36 | E9 EA2EEEFF | JMP 0x13F7EBD25 | Jump near the entrance
```

```
000000013F7EBD18 | 48:895C24 20 | MOV QWORD PTR SS:[RSP+0x20],RBX |
000000013F7EBD1D | 55 | PUSH RBP |
000000013F7EBD1E | 48:8BEC | MOV RBP,RSP |
000000013F7EBD21 | 48:83EC 20 | SUB RSP,0x20 |
000000013F7EBD25 | 48:8B05 84703D00 | MOV RAX,QWORD PTR DS:[0x13FBC2DB0] | Jump here, let’s take a look at the first CALL that is obviously the 64-bit VC code entry
000000013F7EBD2C | 48:8365 18 00 | AND QWORD PTR SS:[RBP+0x18],0x0 |
000000013F7EBD31 | 48:BB 32A2DF2D992B0 | MOV RBX,0x2B992DDFA232 |
000000013F7EBD3B | 48:3BC3 | CMP RAX,RBX |
000000013F7EBD3E | 75 6F | JNE 0x13F7EBDAF |
000000013F7EBD40 | 48:8D4D 18 | LEA RCX,QWORD PTR SS:[RBP+0x18] |
000000013F7EBD44 | FF15 16D81100 | CALL QWORD PTR DS:[<&GetSystemTimeAsFileTime>] |
000000013F7EBD4A | 48:8B45 18 | MOV RAX,QWORD PTR SS:[RBP+0x18] |
000000013F7EBD4E | 48:8945 10 | MOV QWORD PTR SS:[RBP+0x10],RAX |
000000013F7EBD52 | FF15 30DB1100 | CALL QWORD PTR DS:[<&GetCurrentThreadId>] |
000000013F7EBD58 | 8BC0 | MOV EAX,EAX |
000000013F7EBD5A | 48:3145 10 | XOR QWORD PTR SS:[RBP+0x10],RAX |
000000013F7EBD5E | FF15 14DB1100 | CALL QWORD PTR DS:[<&GetCurrentProcessId>] |
000000013F7EBD64 | 48:8D4D 20 | LEA RCX,QWORD PTR SS:[RBP+0x20] |
000000013F7EBD68 | 8BC0 | MOV EAX,EAX |
000000013F7EBD6A | 48:3145 10 | XOR QWORD PTR SS:[RBP+0x10],RAX |
000000013F7EBD6E | FF15 84D71100 | CALL QWORD PTR DS:[<&QueryPerformanceCounter>] |
000000013F7EBD74 | 8B45 20 | MOV EAX,DWORD PTR SS:[RBP+0x20] |
000000013F7EBD77 | 48:C1E0 20 | SHL RAX,0x20 |
000000013F7EBD7B | 48:8D4D 10 | LEA RCX,QWORD PTR SS:[RBP+0x10] |
000000013F7EBD7F | 48:3345 20 | XOR RAX,QWORD PTR SS:[RBP+0x20] |
000000013F7EBD83 | 48:3345 10 | XOR RAX,QWORD PTR SS:[RBP+0x10] |
000000013F7EBD87 | 48:33C1 | XOR RAX,RCX |
000000013F7EBD8A | 48:B9 FFFFFFFFFFFF0 | MOV RCX,0xFFFFFFFFFFFF |
000000013F7EBD94 | 48:23C1 | AND RAX,RCX |
000000013F7EBD97 | 48:B9 33A2DF2D992B0 | MOV RCX,0x2B992DDFA233 |
000000013F7EBDA1 | 48:3BC3 | CMP RAX,RBX |
000000013F7EBDA4 | 48:0F44C1 | CMOVE RAX,RCX |
000000013F7EBDA8 | 48:8905 01703D00 | MOV QWORD PTR DS:[0x13FBC2DB0],RAX |
000000013F7EBDAF | 48:8B5C24 48 | MOV RBX,QWORD PTR SS:[RSP+0x48] |
000000013F7EBDB4 | 48:F7D0 | NOT RAX |
000000013F7EBDB7 | 48:8905 FA6F3D00 | MOV QWORD PTR DS:[0x13FBC2DB8],RAX |
000000013F7EBDBE | 48:83C4 20 | ADD RSP,0x20 |
```

Back here:

```
000000013F7E1075 | 48:83C4 28 | ADD RSP,0x28 | Back here, the bottom line of the first CALL of 64-bit VC
000000013F7E1079 | E9 36FEFFFF | JMP 0x13F7E0EB4 |
```

Refer to the 64-bit VC code entry feature and fill in the code:

```
000000013F7E106C | 48:83EC 28 | SUB RSP,0x28 | Entry address
000000013F7E1070 | E8 A3AC0000 | CALL 0x13F7EBD18 |
000000013F7E1075 | 48:83C4 28 | ADD RSP,0x28 |
000000013F7E1079 | E9 36FEFFFF | JMP 0x13F7E0EB4 |
```

Establish a new operating point at 000000013F7E106C, and then use the Sclla x64 plug-in DUMP to automatically search for IAT, all valid, and repair the DUMP file.

Unplug the dongle, run the program, normal!

carry out!

☆ Collection · 12   Like · 1   ¥ Reward   share it

Latest reply (8)

Engineering 2020-2-19 11:44   2nd floor   0

极客

Cattle

tomtory 2020-2-19 20:18   3rd floor   0

极客

powerful

Li Gang ctt 2020-2-20 15:50   4th floor   0

极客

🙂

It looks like it should be a version above 7.8++, without API padding encryption, SafeNet committed to Gemalto to marry Thales Group and technically regressed, I do not know whether it is "intentional" or "intentional"🙂

mb_xghoecki 2020-2-23 13:24   5th floor   1

临时

Thanks for sharing

Heart Flower Send chen 2020-3-4 09:10   6th floor   0

极客

Thanks for sharing, it's the tutorial I'm looking for

Latest reply (8)

luzhmu 4 2020-3-7 17:08 7th floor 0

极客

Thanks for sharing

wolfing 2 2020-3-25 23:02 8th floor 0

极客

SafeNet shelling thanks for sharing

panman 2 2020-4-5 11:00 9th floor 0

极客

Admire, learn.... I can't do this blasting

Tourist

Login | Register to reply

Reply expression Advanced reply

return