

>> 抓病毒行为：重点是 exchange1.dll 这个释放文件；
oledump.py 尝试获取一下 office 文件中的宏代码；
https://blog.csdn.net/Cody_Ren/article/details/79941643

>> 对其脱壳：

push ebp 之后 esp 定律

Address	Disassembly	Comment
00401C30	push ebp	EntryPoint
00401C31	mov ebp, esp	
00401C33	sub esp, 94	
00401C39	mov dword ptr ds:[411990], ebx	
00401C3F	mov dword ptr ds:[411980], edi	
00401C45	mov dword ptr ds:[41198C], esi	
00401C4B	mov dword ptr ds:[411984], ebp	
00401C51	mov dword ptr ss:[ebp-10], F0F6E4E2	
00401C58	lea eax, dword ptr ss:[ebp-10]	
00401C5B	mov dword ptr ss:[ebp-4], eax	
00401C5E	mov eax, dword ptr ss:[ebp-10]	
00401C61	or eax, 7660	
00401C66	mov ecx, dword ptr ss:[ebp-10]	
00401C69	add ecx, 1	
00401C6C	cdq	
00401C6D	idiv ecx	
00401C6F	mov dword ptr ss:[ebp-8], eax	
00401C72	push 0	
00401C74	push 0	
00401C76	call dword ptr ds:[<&SetClipboardData>]	
00401C7C	call dword ptr ds:[<&GetLastError>]	

go 一下：

Address	Disassembly
000E04E8	jmp edx
000E04EA	xor eax, eax
000E04EC	pop edi
000E04ED	pop esi
000E04EE	pop ebx
000E04EF	mov esp, ebp
000E04F1	pop ebp

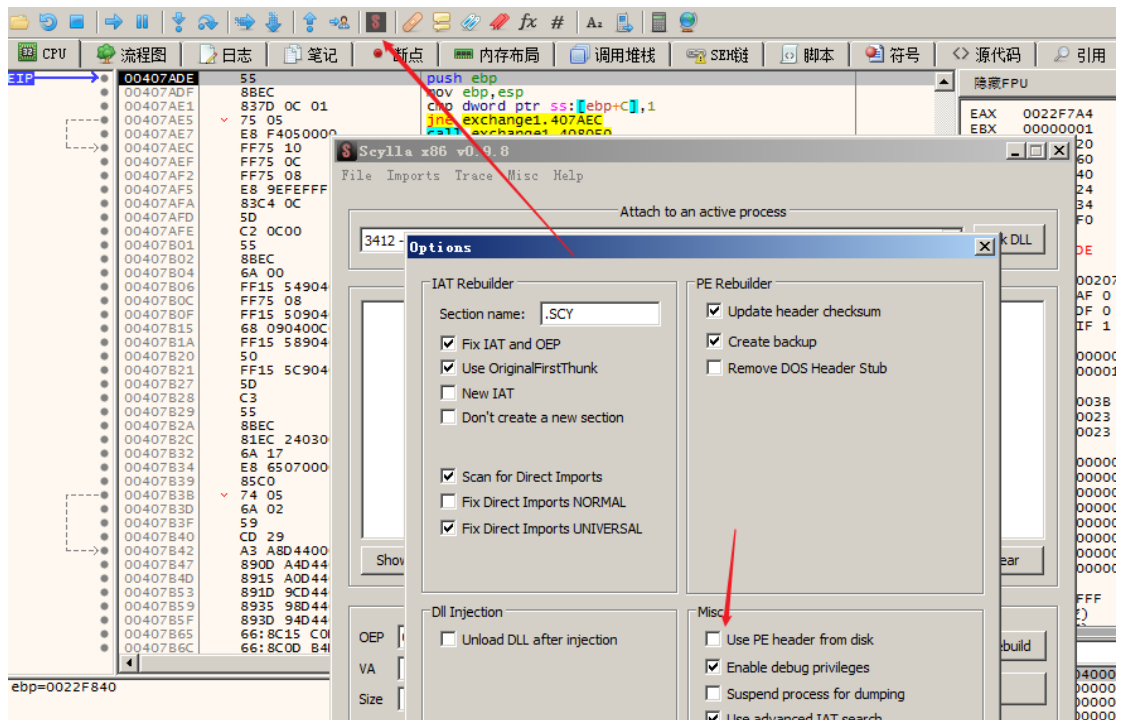
pushad 后再次 esp 定律

Address	Disassembly
00414660	cmp byte ptr ss:[esp+8], 1
00414665	jne exchange1.414824
0041466B	pushad
0041466C	mov esi, exchange1.40F000
00414671	lea edi, dword ptr ds:[esi-E000]
00414677	push edi
00414678	jmp exchange1.41468A
0041467A	nop
0041467B	nop

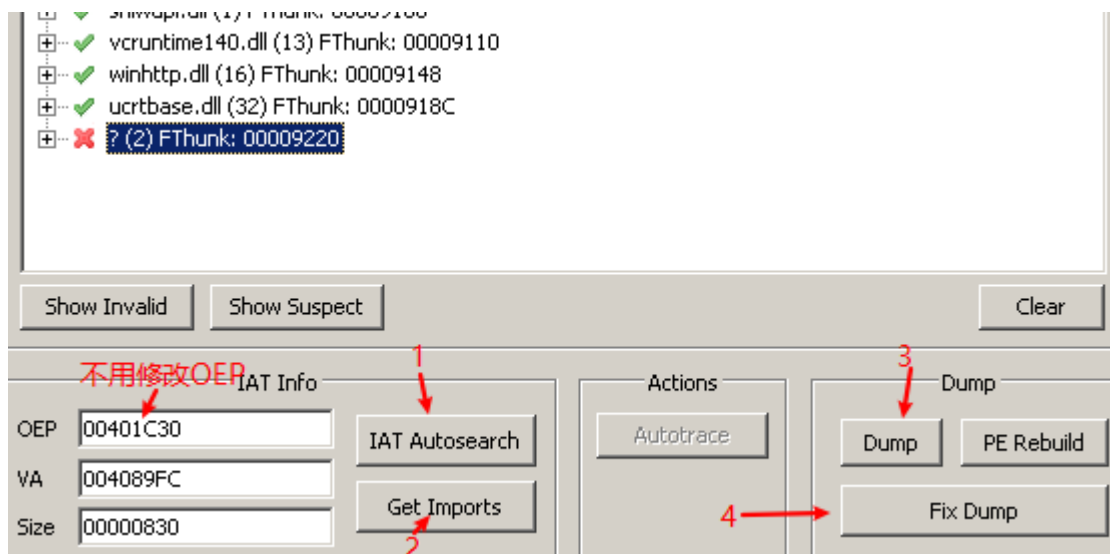
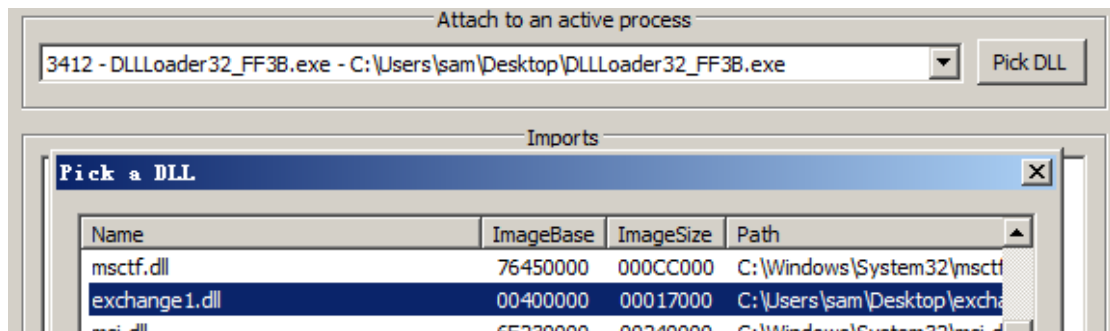
go

Address	Disassembly
00414817	lea eax, dword ptr ss:[esp-80]
0041481B	push 0
0041481D	cmp esp, eax
0041481F	jne exchange1.414818
00414821	sub esp, FFFFFFFF80
00414824	jmp exchange1.407ADE
00414829	add byte ptr ds:[eax], al
0041482B	add byte ptr ds:[eax+ecx*2+41], al
0041482F	add byte ptr ds:[eax+ecx*2+41], cl

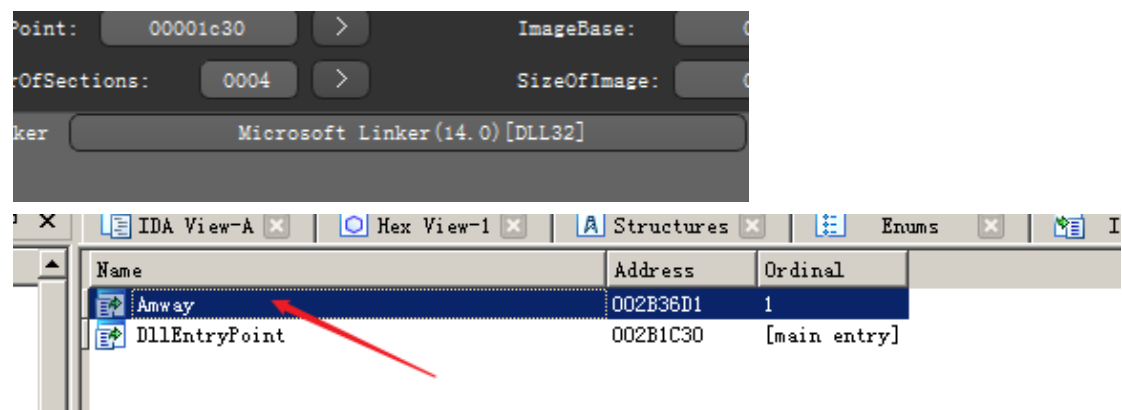
jmp 跳过去，开始 dUmp，先设置一下：



pickDLL 选择



脱壳完毕，静态分析即可。



>> 也可以带壳调试:

根据 bp VirtualAlloc 下断, 不断 F9 反汇编观察返回到。来自的地址, 查看参数地址内容即可。

<https://www.pianshen.com/article/32721181205/>

