

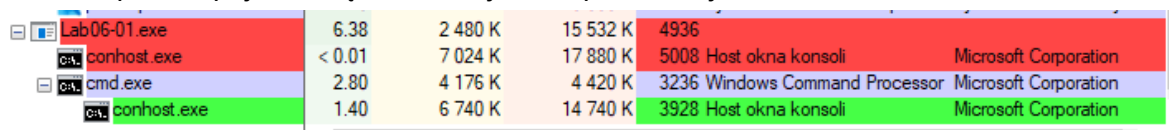
Zaawansowana analiza dynamiczna

Laboratorium 5.1

Przeprowadź analizę pliku Lab06-01.exe za pomocą programu OllyDbg i odpowiedz na poniższe pytania:

1. W jaki sposób można zmusić malware do instalacji?

Po próbie uruchomienia programu z poziomu eksploratora systemu plików Windows - plik wykonywalny znika. Również podobnie w przypadku programu `process explorer` - pierwotnie proces pojawia się na liście, jednak po krótkiej chwili znika



Lab06-01.exe	6.38	2 480 K	15 532 K	4936		
conhost.exe	< 0.01	7 024 K	17 880 K	5008	Host okna konsoli	Microsoft Corporation
cmd.exe	2.80	4 176 K	4 420 K	3236	Windows Command Processor	Microsoft Corporation
conhost.exe	1.40	6 740 K	14 740 K	3928	Host okna konsoli	Microsoft Corporation

Uruchamia on proces konsoli `cmd.exe`

Następnie uruchomiłem program `ollydbg` i starałem się znaleźć czegoś związanego z konsolą właśnie

Address	Hex dump	ASCII
0040C0A0	72 62 00 00 60 00 00 00	rb..*...
0040C0A8	43 4D 44 00 44 4F 57 4E	CMD.DOWN
0040C0B0	4C 4F 41 44 00 00 00 00	LOAD....
0040C0B8	55 50 4C 4F 41 44 00 00	UPLOAD..
0040C0C0	20 00 00 00 53 4C 45 45	...SLEE
0040C0C8	50 00 00 00 63 6D 64 2E	P...cmd.
0040C0D0	65 78 65 00 20 3E 3E 20	exe. >>
0040C0D8	4E 55 4C 00 2F 63 20 64	NUL./c d
0040C0E0	65 6C 20 00 75 70 73 00	el .ups.
0040C0E8	68 74 74 70 3A 2F 2F 77	http://w
0040C0F0	77 77 2E 70 72 61 63 74	ww.pract
0040C0F8	69 63 61 6C 6D 61 6C 77	icalmalw
0040C100	61 72 65 61 6E 61 6C 79	areanaly
0040C108	73 69 73 2E 63 6F 6D 00	sis.com.
0040C110	38 30 00 00 36 30 00 00	80..60..
0040C118	3A 4D 21 2E 21 27 2E 73	Ma...xx

Po analizie sąsiednich adresów można zauważyć, że uruchamia się `Service Manager` odpowiedzialny za instalację.

Address	Hex dump	ASCII
0040C0C0	20 00 00 00 53 4C 45 45	...SLEE
0040C0C8	50 00 00 00 63 60 64 2E	P...cmd.
0040C0D0	65 78 65 00 20 3E 3E 20	exe. >>
0040C0D8	4E 55 4C 00 2F 63 20 64	NUL./c d
0040C0E0	65 6C 20 00 75 70 73 00	el .ups.
0040C0E8	68 74 74 70 3A 2F 2F 77	http://w
0040C0F0	77 77 2E 70 72 61 63 74	ww.pract
0040C0F8	69 63 61 6C 6D 61 6C 77	icalmalw
0040C100	61 72 65 61 6E 61 6C 79	areanaly
0040C108	73 69 73 2E 63 6F 6D 00	sis.com.
0040C110	38 30 00 00 36 30 00 00	80..60..
0040C118	20 40 61 6E 61 67 65 72	Manager
0040C120	20 53 65 72 76 69 63 65	Service
0040C128	00 00 00 00 2E 65 78 65exe
0040C130	00 00 00 00 25 53 59 53%SYS
0040C138	54 45 4D 52 4F 4F 54 25	TEMROOT%
0040C140	5C 73 79 73 74 65 6D 33	\system3
0040C148	32 5C 00 00 68 3A 25 73	2\..k;%s
0040C150	20 68 3A 25 73 20 70 3A	h;%s p:
0040C158	25 73 20 70 65 72 3A 25	%s per;%
0040C160	73 0A 00 00 2D 63 63 00	s...-cc.
0040C168	2D 63 00 00 2D 72 65 00	-c...-re.
0040C170	2D 69 6E 00 00 00 00 00	-in.....
0040C178	00 00 00 00 00 00 00 00
0040C180	01 00 00 00 00 00 00 00	0.....
0040C188	00 00 00 00 00 00 00 00
0040C190	BA 2D 40 00 01 00 00 00	-@.0...
0040C198	00 B2 40 00 F0 B1 40 00	000.-000.
0040C1A0	A0 F1 40 00 00 00 00 00	á@.....
0040C1A8	A0 F1 40 00 01 01 00 00	á@.00...
0040C1B0	00 00 00 00 00 00 00 00
0040C1B8	00 10 00 00 00 00 00 00	.>.....

Niestety nie jestem w stanie sprawdzić w jaki dokładnie sposób wymusić instalację złośliwego oprogramowania

Jedynie co mi wiadomo to, że podczas instalacji próbuje zmienić coś w rejestrze oraz tworzyć pliki - kopiuje pliki do `System32`

CPU - main thread, module Lab06-01			
0040120E	CC	INT3	pDisposition = NULL pHandle pSecurity = NULL Access = KEY_ALL_ACCESS Options = REG_OPTION_NON_VOLATILE Class = NULL Reserved = 0 Subkey = "SOFTWARE\Microsoft \XPS" hKey = HKEY_LOCAL_MACHINE RegCreateKeyExA
0040120F	CC	INT3	
00401210	55	PUSH EBP	
00401211	8BEC	MOV EBP,ESP	
00401213	83EC 08	SUB ESP,8	ValueName = "Configuration" hKey RegDeleteValueA
00401216	6A 00	PUSH 0	
00401218	8D45 F8	LEA EAX,DWORD PTR SS:[EBP-8]	
0040121B	50	PUSH EAX	
0040121C	6A 00	PUSH 0	hObject CloseHandle
0040121E	68 3F00F00	PUSH 0F003F	
00401223	6A 00	PUSH 0	
00401225	6A 00	PUSH 0	
00401227	6A 00	PUSH 0	hObject CloseHandle
00401229	68 40C04000	PUSH Lab06-01.0040C040	
0040122E	68 02000000	PUSH 80000002	
00401233	FF15 18B04000	CALL DWORD PTR DS:[<&ADVAPI32.RegCreate	
00401239	85C0	TEST EAX,EAX	hObject CloseHandle
0040123B	74 07	JE SHORT Lab06-01.00401244	
0040123D	B8 01000000	MOV EAX,1	
00401242	EB 35	JMP SHORT Lab06-01.00401279	
00401244	68 30C04000	PUSH Lab06-01.0040C030	hObject CloseHandle
00401249	8B4D F8	MOV ECX,DWORD PTR SS:[EBP-8]	
0040124C	51	PUSH ECX	
0040124D	FF15 14B04000	CALL DWORD PTR DS:[<&ADVAPI32.RegDelete	
00401253	8945 FC	MOV DWORD PTR SS:[EBP-4],EAX	hObject CloseHandle
00401256	837D FC 00	CMP DWORD PTR SS:[EBP-4],0	
0040125A	74 11	JE SHORT Lab06-01.0040126D	
0040125C	8B55 F8	MOV EDX,DWORD PTR SS:[EBP-8]	
0040125F	52	PUSH EDX	hObject CloseHandle
00401260	FF15 64B04000	CALL DWORD PTR DS:[<&KERNEL32.CloseHand	
00401266	B8 01000000	MOV EAX,1	
0040126B	EB 0C	JMP SHORT Lab06-01.00401279	
0040126D	8B45 F8	MOV EAX,DWORD PTR SS:[EBP-8]	hObject CloseHandle
00401270	50	PUSH EAX	
00401271	FF15 64B04000	CALL DWORD PTR DS:[<&KERNEL32.CloseHand	
00401277	33C0	XOR EAX,EAX	

Oraz uruchamia kolejne procesy .exe

2. Podaj argumenty wiersza poleceń dla tego programu. Jakie są wymagania dotyczące hasła?

Zważając na powyższą konstrukcję stringów, można przypuszczać, że po uruchomieniu `cmd` program przyjmuje następujące argumenty

- `-cc`
- `-c`
- `-re`
- `-in`

Address	Hex dump	ASCII
0040C0C0	20 00 00 00 53 4C 45 45	...SLEE
0040C0C8	50 00 00 00 63 60 64 2E	P...cmd.
0040C0D0	65 78 65 00 20 3E 3E 20	exe. >>
0040C0D8	4E 55 4C 00 2F 63 20 64	NUL./c d
0040C0E0	65 6C 20 00 75 70 73 00	el .ups.
0040C0E8	68 74 74 70 3A 2F 2F 77	http://w
0040C0F0	77 77 2E 70 72 61 63 74	ww.pract
0040C0F8	69 63 61 6C 6D 61 6C 77	icalmalw
0040C100	61 72 65 61 6E 61 6C 79	areanaly
0040C108	73 69 73 2E 63 6F 6D 00	sis.com.
0040C110	38 30 00 00 36 30 00 00	80..60..
0040C118	20 4D 61 6E 61 67 65 72	Manager
0040C120	20 53 65 72 76 69 63 65	Service
0040C128	00 00 00 00 2E 65 78 65exe
0040C130	00 00 00 00 25 53 59 53	...%SYS
0040C138	54 45 4D 52 4F 4F 54 25	TEMROOT%
0040C140	5C 73 79 73 74 65 6D 33	\system3
0040C148	32 5C 00 00 68 3A 25 73	2\..k:%s
0040C150	20 68 3A 25 73 20 70 3A	h:%s p:
0040C158	25 73 20 70 65 72 3A 25	%s per:%
0040C160	73 0A 00 00 2D 63 63 00	s...-cc.
0040C168	2D 63 00 00 2D 72 65 00	-c...-re.
0040C170	2D 69 6E 00 00 00 00 00	-in.....
0040C178	00 00 00 00 00 00 00 00
0040C180	01 00 00 00 00 00 00 00	0.....
0040C188	00 00 00 00 00 00 00 00
0040C190	BA 2D 40 00 01 00 00 00	-@.0...
0040C198	00 B2 40 00 F0 B1 40 00	.000.-000.
0040C1A0	A0 F1 40 00 00 00 00 00	á"0.00...
0040C1A8	A0 F1 40 00 01 01 00 00	á"0.00...
0040C1B0	00 00 00 00 00 00 00 00
0040C1B8	00 10 00 00 00 00 00 00	..>.....

Następnie trochę wspomogłem się programem `IDA`

.text:00402B3A	call	sub_402410
.text:00402B3F		
.text:00402B3F loc_402B3F:		; CODE XREF: sub_402AF0+48↑j
.text:00402B3F	mov	ecx, [ebp+0Ch]
.text:00402B42	mov	edx, [ecx+4]
.text:00402B45	mov	[ebp-1820h], edx
.text:00402B48	push	offset aIn ; "-in"
.text:00402B50	mov	eax, [ebp-1820h]
.text:00402B56	push	eax
.text:00402B57	call	sub_40380F
.text:00402B5C	add	esp, 8
.text:00402B5F	test	eax, eax
.text:00402B61	jnz	short loc_402BC7
.text:00402B63	cmp	dword ptr [ebp+8], 3
.text:00402B67	jnz	short loc_402B9A
.text:00402B69	push	400h
.text:00402B6E	lea	ecx, [ebp-404h]
.text:00402B74	push	ecx
.text:00402B75	call	sub_4025B0
.text:00402B7A	add	esp, 8
.text:00402B7D	test	eax, eax
.text:00402B7F	jz	short loc_402B89
.text:00402B81	or	eax, 0FFFFFFFh
.text:00402B84	jmp	loc_402D78
.text:00402B89 ;		

```

.text:004025B0                                     ; sub_402900+CB↓p ...
.text:004025B0
.text:004025B0 var_400 = byte ptr -400h
.text:004025B0
.text:004025B0 push ebp
.text:004025B1 mov ebp, esp
.text:004025B3 sub esp, 400h
.text:004025B9 push 400h ; nSize
.text:004025BE lea eax, [ebp+var_400]
.text:004025C4 push eax ; lpFilename
.text:004025C5 push 0 ; hModule
.text:004025C7 call ds:GetModuleFileNameA
.text:004025CD test eax, eax
.text:004025CF jnz short loc_4025D8
.text:004025D1 mov eax, 1
.text:004025D6 jmp short loc_4025F3
.text:004025D8 ; -----

```

Udało mi się znaleźć funkcję odpowiedzialną za przyjmowanie parametru `-in` (który najprawdopodobniej odpowiada za instalację właśnie)

```

; Attributes: bp-based frame

sub_402510 proc near
push    ebp
mov     ebp, esp
push    ecx
push    edi
mov     edi, [ebp+8]
or      ecx, 0FFFFFFFh
xor     eax, eax
repne scasb
not     ecx
add     ecx, 0FFFFFFFh
cmp     ecx, 4
jz      short loc_40252D

```

Idąc dalej można zobaczyć że jednymi z liter hasła jest **a** oraz **c**

```
.....  
:ax  
  
FFFFFFFFh  
loc_40252D
```

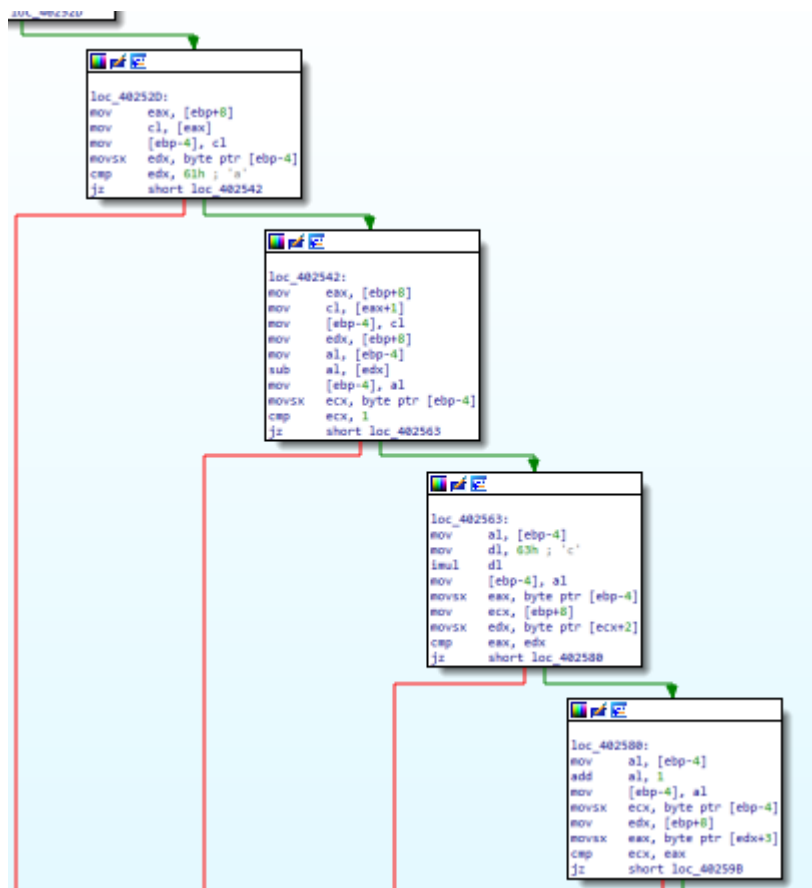
```
loc_40252D:  
mov     eax, [ebp+8]  
mov     cl, [eax]  
mov     [ebp-4], cl  
movsx   edx, byte ptr [ebp-4]  
cmp     edx, 61h ; 'a'  
jz      short loc_402542
```

loc_402542

```
dx]  
], al  
/te ptr [ebp-4]  
loc_402563
```

```
loc_402563:  
mov     al, [ebp-4]  
mov     dl, 63h ; 'c'  
imul    dl  
mov     [ebp-4], al  
movsx   eax, byte ptr [ebp-4]  
mov     ecx, [ebp+8]  
movsx   edx, byte ptr [ecx+2]  
cmp     eax, edx  
jz      short loc_402580
```





Każda kolejna litera hasła wywołuje funkcję odpowiadającą za kolejną literę - pierwszą jest **a**, następna jest o inkrementowana, więc będzie to litera **b**, litera **c** jest podana, a czwarta jest znowu o jeden większa od poprzedniej - **d**
 W takim razie hasło będzie miało następującą postać - **abcd**

3. Jak można wykorzystać OllyDbg do wprowadzenia zmian w tym malware, aby nie wymagał podawania hasła w wierszu poleceń?

Należy nadpisać poszczególne funkcje (wcześniej odnalezione w programie IDA) odpowiedzialne za zwracanie informacji o tym iż sprawdzanie hasła nie zakończyło się sukcesem

Czyli w tym miejscu należy zmusić program do zwracania prawdy

0040250F	CC	INT3
00402510	55	PUSH EBP
00402511	8BEC	MOV EBP,ESP
00402513	51	PUSH ECX
00402514	57	PUSH EDI
00402515	8B7D 08	MOV EDI,DWORD PTR SS:[EBP+8]
00402518	83C9 FF	OR ECX,FFFFFFFF
00402518	33C0	XOR EAX,EAX
0040251D	F2:AE	REPNE SCAS BYTE PTR ES:[EDI]
0040251F	F7D1	NOT ECX
00402521	83C1 FF	ADD ECX,-1
00402524	83F9 04	CMP ECX,4

Można to zrobić używając funkcji `mov eax,1` oraz `ret`

0040250F	CC	INT3
00402510	B8 01000000	MOV EAX,1
00402515	C3	RETN
00402516	90	NOP
00402517	90	NOP
00402518	. 83C9 FF	OR ECX,FFFFFFFF
0040251B	. 33C0	XOR EAX,EAX
0040251D	. F2:AE	REPNE SCAS BYTE PTR ES:[EDI]
0040251F	. F7D1	NOT ECX
00402521	. 83C1 FF	ADD ECX,-1
00402524	. 83F9 04	CMP ECX,4
00402527	. 74 04	JE SHORT Lab06-01.0040252D

4. Podaj indykatory hostowe związane z tym malware.

Tak jak w poprzednich plikach - www.practicalmalwareanalysis.com

0040C0E8	88 74 74 70 5H 2F 2F 7C	http://w
0040C0F0	77 77 2E 70 72 61 63 74	ww.pract
0040C0F8	69 63 61 6C 6D 61 6C 77	icalmalw
0040C100	61 72 65 61 6E 61 6C 79	areanaly
0040C108	73 69 73 2E 63 6F 6D 00	sis.com.
0040C110	38 30 00 00 36 30 00 00	80..60..
0040C118	20 4D 61 6F 61 67 65 72	Managew

004028C5	. B8 01000000	MOV EAX,1	Arg4 = 0040C114 ASCII "60"
004028CA	> EB 29	JMP SHORT Lab06-01.004028F5	Arg3 = 0040C110 ASCII "80"
004028CC	. 68 14C14000	PUSH Lab06-01.0040C114	Arg2 = 0040C0E8 ASCII "http://www.practicalmalwareanalysis.com"
004028D1	. 68 10C14000	PUSH Lab06-01.0040C110	Arg1 = 0040C0E4 ASCII "ups"
004028D6	. 68 E8C04000	PUSH Lab06-01.0040C0E8	Lab06-01.00401070
004028DB	. 68 E4C04000	PUSH Lab06-01.0040C0E4	
004028E0	. E8 8BE7FFFF	CALL Lab06-01.00401070	
004028E5	. 83C4 10	ADD ESP,10	
004028F1	. 9FC0	TEST EAX,EAX	

oraz najprawdopodobniej numery portów

- 60
- 80

5. Opisz działania umożliwiające wykorzystanie złośliwego pliku przy pomocy sieci Internet.

0040C080	47 45 54 20 00 00 00 00	GET
0040C088	27 60 27 60 27 00 00 00
0040C090	60 27 60 27 60 00 00 00
0040C098	4E 4F 54 48 49 4E 47 00	NOTHING.
0040C0A0	72 62 00 00 60 00 00 00	rb.....
0040C0A8	43 40 44 00 44 4F 57 4E	CMD.DOWN
0040C0B0	4C 4F 41 44 00 00 00 00	LOAD....
0040C0B8	55 50 4C 4F 41 44 00 00	UPLOAD..
0040C0C0	20 00 00 00 53 4C 45 45	...SLEE
0040C0C8	50 00 00 00 63 6D 64 2E	P...cmd.
0040C0D0	65 78 65 00 20 3E 3E 20	exe. >>

Program może pobierać coś z sieci (**DOWNLOAD**), uploadować (**UPLOAD**), czekać (**SLEEP**), prawdopodobnie uruchamiać komendę w konsoli (**CMD**) oraz nie robić nic (**NOTHING**)

6. Wymień przydatne wskaźniki sieciowe tego malware.

004028C5	. B8 01000000	MOV EAX,1	Arg4 = 0040C114 ASCII "60"
004028CA	> EB 29	JMP SHORT Lab06-01.004028F5	Arg3 = 0040C110 ASCII "80"
004028CC	. 68 14C14000	PUSH Lab06-01.0040C114	Arg2 = 0040C0E8 ASCII "http://www.practicalmalwareanalysis.com"
004028D1	. 68 10C14000	PUSH Lab06-01.0040C110	Arg1 = 0040C0E4 ASCII "ups"
004028D6	. 68 E8C04000	PUSH Lab06-01.0040C0E8	Lab06-01.00401070
004028DB	. 68 E4C04000	PUSH Lab06-01.0040C0E4	
004028E0	. E8 8BE7FFFF	CALL Lab06-01.00401070	
004028E5	. 83C4 10	ADD ESP,10	
004028F1	. 9FC0	TEST EAX,EAX	

Link URL, numery portów

Laboratorium 5.2

Przeprowadź analizę pliku Lab06-02.exe za pomocą programu OllyDbg i odpowiedz na poniższe pytania:

1. Przeanalizuj i wypisz łańcuchy znaków, które jesteśmy w stanie odszukać w pliku.

Są to głównie napisy z importowanych bibliotek, nazwy funkcji i modułów oraz kody błędów

WaitForSingleObject	HeapAlloc
GetStartupInfo	VirtualAlloc
WS2_32.dll	HeapReAlloc
HeapDestroy	GetStringType
HeapCreate	GetStringType
VirtualFree	GetFileType
HeapFree	GetCommandLine
HeapAlloc	ExitProcess
VirtualAlloc	GetCurrentProcess
HeapReAlloc	FreeEnvironmentStrings
GetStringType	FreeEnvironmentStrings
GetStringType	Sleep
GetFileType	UnhandledExceptionFilter
GetCommandLine	GetModuleFileName
ExitProcess	GetProcAddress
GetCurrentProcess	LoadLibrary
FreeEnvironmentStrings	cmd
FreeEnvironmentStrings	GetVersion
Sleep	WideCharToMultiByte
UnhandledExceptionFilter	SetHandleCount
GetModuleFileName	GetStdHandle
GetProcAddress	RtlUnwind
LoadLibrary	GetCPIInfo
cmd	GetOEMCP
GetVersion	MultiByteToWideChar
WideCharToMultiByte	LCMapString
SetHandleCount	LCMapString
GetStdHandle	user32.dll
RtlUnwind	KERNEL32.dll
GetCPIInfo	!This program cannot be run in DOS mode.
GetOEMCP	-

encoding (2)	size (bytes)	location	flag (6)	label (42)	group (9)	technique (5)	value (352)
ascii	9	0x00004572	x	import	network	-	WSASocket
ascii	9	0x000046FE	x	import	file	-	WriteFile
ascii	13	0x00004534	x	import	execution	Execution through API	CreateProcess
ascii	16	0x000045BA	x	import	execution	-	TerminateProcess
ascii	21	0x00004648	x	import	execution	-	GetEnvironmentStrings
ascii	21	0x00004660	x	import	execution	-	GetEnvironmentStrings

Z podejrzanych rzeczy udało się znaleźć funkcje odpowiedzialne za pisanie do pliku, zarządzanie procesami i socketami

2. Opisz wynik działania z uruchomienia tego pliku.

Program się nie uruchamia. Nie widać żadnej aktywności nawet po sprawdzeniu w eksploratorze procesów. Być może program odmawia uruchomienia się z uwagi na to, że wykrywa że jest na wirtualnej maszynie

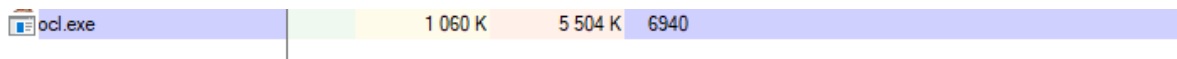
Nie wiem czy jest to przypadek, ale czasem po uruchomieniu pliku uruchamia się również proces `svchost.exe` - być może malware próbuje wykonać jakieś połączenie internetowe

3. W jaki sposób zmusić analizowany plik do uruchomienia swojej szkodliwej zawartości?

Próbowałem znaleźć informacji z użyciem `OlllyDBG` aczkolwiek nie udało mi się znaleźć niczego pomocnego.

Próbowałem również z użyciem programu IDA aczkolwiek pierwotnie nie udało mi się nic znaleźć.

DO dopiero po wykonaniu następnego podpunktu udało mi się uruchomić program -> należy go nazwać `ocl.exe`



4. Opisz działania znajdujące się pod adresem 0x00401133.

00401133	• 57	PUSH EDI
00401138	• C685 50FFFFFF	MOV BYTE PTR SS:[EBP-1B0],31
0040113A	• C685 51FFFFFF	MOV BYTE PTR SS:[EBP-1AF],71
00401141	• C685 52FFFFFF	MOV BYTE PTR SS:[EBP-1AE],61
00401148	• C685 53FFFFFF	MOV BYTE PTR SS:[EBP-1AD],7A
0040114F	• C685 54FFFFFF	MOV BYTE PTR SS:[EBP-1AC],32
00401156	• C685 55FFFFFF	MOV BYTE PTR SS:[EBP-1AB],77
0040115D	• C685 56FFFFFF	MOV BYTE PTR SS:[EBP-1AA],73
00401164	• C685 57FFFFFF	MOV BYTE PTR SS:[EBP-1A9],78
0040116B	• C685 58FFFFFF	MOV BYTE PTR SS:[EBP-1A8],33
00401172	• C685 59FFFFFF	MOV BYTE PTR SS:[EBP-1A7],65
00401179	• C685 5AFFFFFF	MOV BYTE PTR SS:[EBP-1A6],64
00401180	• C685 5BFFFFFF	MOV BYTE PTR SS:[EBP-1A5],63
00401187	• C685 5CFFFFFF	MOV BYTE PTR SS:[EBP-1A4],0
0040118E	• C685 60FFFFFF	MOV BYTE PTR SS:[EBP-1A0],6F
00401195	• C685 61FFFFFF	MOV BYTE PTR SS:[EBP-19F],63
0040119C	• C685 62FFFFFF	MOV BYTE PTR SS:[EBP-19E],6C
004011A3	• C685 63FFFFFF	MOV BYTE PTR SS:[EBP-19D],2E
004011AA	• C685 64FFFFFF	MOV BYTE PTR SS:[EBP-19C],65
004011B1	• C685 65FFFFFF	MOV BYTE PTR SS:[EBP-19B],78
004011B8	• C685 66FFFFFF	MOV BYTE PTR SS:[EBP-19A],65
004011BF	• C685 67FFFFFF	MOV BYTE PTR SS:[EBP-199],0

Jest to ładowanie pojedynczych znaków ASCII na stos o następującej postaci:

- **1qaz2wsx3edc**
- **ocl.exe**

5. Podaj argumenty, które są przekazywane do podprogramu pod adresem 0x00401089?

00401089	55	PUSH EBP
0040108A	8BEC	MOV EBP,ESP
0040108C	81EC 08010000	SUB ESP,108
00401092	57	PUSH EDI
00401093	C785 F8FEFFFF	MOV DWORD PTR SS:[EBP-108],0
0040109D	C685 00FFFFFF	MOV BYTE PTR SS:[EBP-100],0
004010A4	B9 3F000000	MOV ECX,3F
004010A9	33C0	XOR EAX,EAX
004010AB	8DB0 01FFFFFF	LEA EDI,DWORD PTR SS:[EBP-FF]
004010B1	F3:AB	REP STOS DWORD PTR ES:[EDI]
004010B3	66:AB	STOS WORD PTR ES:[EDI]
004010B5	AA	STOS BYTE PTR ES:[EDI]
004010B6	8B45 08	MOV EAX,DWORD PTR SS:[EBP+8]
004010B9	50	PUSH EAX
004010BA	E8 81030000	CALL Lab06-02.00401440
004010BF	83C4 04	ADD ESP,4
004010C2	8985 FCFEFFFF	MOV DWORD PTR SS:[EBP-104],EAX
004010C8	C785 F8FEFFFF	MOV DWORD PTR SS:[EBP-108],0
004010D2	EB 0F	JMP SHORT Lab06-02.004010E3
004010D4	8B8D F8FEFFFF	MOV ECX,DWORD PTR SS:[EBP-108]
004010DA	83C1 01	ADD ECX,1
004010DD	898D F8FEFFFF	MOV DWORD PTR SS:[EBP-108],ECX
004010E3	83BD F8FEFFFF	CMP DWORD PTR SS:[EBP-108],20
004010EA	7D 31	JGE SHORT Lab06-02.0040111D
004010EC	8B55 0C	MOV EDX,DWORD PTR SS:[EBP+C]
004010EF	0395 F8FEFFFF	ADD EDX,DWORD PTR SS:[EBP-108]
004010F5	0FB00A	MOVSX ECX,BYTE PTR DS:[EDX]
004010F8	8B85 F8FEFFFF	MOV EAX,DWORD PTR SS:[EBP-108]
004010FE	99	CDQ
004010FF	F7BD FCFEFFFF	IDIV DWORD PTR SS:[EBP-104]
00401105	8B45 08	MOV EAX,DWORD PTR SS:[EBP+8]
00401108	0FB01410	MOVSX EDX,BYTE PTR DS:[EAX+EDX]
0040110C	33CA	XOR ECX,EDX
0040110E	8B85 F8FEFFFF	MOV EAX,DWORD PTR SS:[EBP-108]
00401114	8B8C05 00FFFFFF	MOV BYTE PTR SS:[EBP+EAX-100],CL
00401118	EB B7	JMP SHORT Lab06-02.004010D4
0040111D	8D85 00FFFFFF	LEA EAX,DWORD PTR SS:[EBP-100]
00401123	5F	POP EDI

są to poprzednio znalezione stringi

6. Podaj nazwę domeny, która wykorzystuje ten malware.

Standardowo www.practicalmalwareanalysis.com

7. Jaka procedura kodowania została zastosowana przez ten program do zaciemnienia nazwy domeny?

XOR

004010C8	C785 F8FEFFFF	MOV DWORD PTR SS:[EBP-108],0
004010D2	EB 0F	JMP SHORT Lab06-02.004010E3
004010D4	8B8D F8FEFFFF	MOV ECX,DWORD PTR SS:[EBP-108]
004010DA	83C1 01	ADD ECX,1
004010DD	898D F8FEFFFF	MOV DWORD PTR SS:[EBP-108],ECX
004010E3	83BD F8FEFFFF	CMP DWORD PTR SS:[EBP-108],20
004010EA	7D 31	JGE SHORT Lab06-02.0040111D
004010EC	8B55 0C	MOV EDX,DWORD PTR SS:[EBP+C]
004010EF	0395 F8FEFFFF	ADD EDX,DWORD PTR SS:[EBP-108]
004010F5	0FB00A	MOVSX ECX,BYTE PTR DS:[EDX]
004010F8	8B85 F8FEFFFF	MOV EAX,DWORD PTR SS:[EBP-108]
004010FE	99	CDQ
004010FF	F7BD FCFEFFFF	IDIV DWORD PTR SS:[EBP-104]
00401105	8B45 08	MOV EAX,DWORD PTR SS:[EBP+8]
00401108	0FB01410	MOVSX EDX,BYTE PTR DS:[EAX+EDX]
0040110C	33CA	XOR ECX,EDX
0040110E	8B85 F8FEFFFF	MOV EAX,DWORD PTR SS:[EBP-108]
00401114	8B8C05 00FFFFFF	MOV BYTE PTR SS:[EBP+EAX-100],CL
00401118	EB B7	JMP SHORT Lab06-02.004010D4
0040111D	8D85 00FFFFFF	LEA EAX,DWORD PTR SS:[EBP-100]
00401123	5F	POP EDI
00401124	8B55	MOV ESP,EBP
00401126	5D	POP EBP
00401127	C3	RET
00401128	55	PUSH EBP
00401129	8BEC	MOV EBP,ESP
0040112B	81EC 04030000	SUB ESP,304

8. Opisz znaczenie wywołania CreateProcessA znajdującego się pod adresem 0x0040106E w nawiązaniu do tego malware?

0040104A	. 8945 E8	MOV DWORD PTR SS:[EBP-18],EAX	
0040104D	. 8B4D E8	MOV ECX,DWORD PTR SS:[EBP-18]	
00401050	. 894D E4	MOV DWORD PTR SS:[EBP-1C],ECX	
00401053	. 8D55 F0	LEA EDX,DWORD PTR SS:[EBP-10]	
00401056	. 52	PUSH EDX	
00401057	. 8D45 A8	LEA EAX,DWORD PTR SS:[EBP-58]	pProcessInfo
0040105A	. 50	PUSH EAX	pStartupInfo
0040105B	. 6A 00	PUSH 0	CurrentDir = NULL
0040105D	. 6A 00	PUSH 0	pEnvironment = NULL
0040105F	. 6A 00	PUSH 0	CreationFlags = 0
00401061	. 6A 01	PUSH 1	InheritHandles = TRUE
00401063	. 6A 00	PUSH 0	pThreadSecurity = NULL
00401065	. 6A 00	PUSH 0	pProcessSecurity = NULL
00401067	. 68 30504000	PUSH Lab06-02.00405030	CommandLine = "cmd"
0040106C	. 6A 00	PUSH 0	ModuleFileName = NULL
0040106E	. FF15 04404000	CALL DWORD PTR DS:[<&KERNEL32.CreateProcessA]	CreateProcessA
00401074	. 8945 EC	MOV DWORD PTR SS:[EBP-14],EAX	
00401077	. 6A FF	PUSH -1	Timeout = INFINITE
00401079	. 8B4D F0	MOV ECX,DWORD PTR SS:[EBP-10]	
0040107C	. 51	PUSH ECX	
0040107D	. FF15 00404000	CALL DWORD PTR DS:[<&KERNEL32.WaitForSingleObject]	WaitForSingleObject
00401083	. 33C0	XOR EAX,EAX	
00401085	. 8BE5	MOV ESP,EBP	

Wykonuje on komendy wiersza poleceń **cmd**

Laboratorium 5.3

Przeprowadź analizę pliku Lab06-03.exe za pomocą programu OllyDbg i IDA. Ten malware ładuje dodatkowe 3 biblioteki DLL (DLL1.dll, DLL2.dll i DLL3.dll), które muszą znajdować się w tej samej lokalizacji podczas ładowania do pamięci. Podczas przeglądania tych bibliotek DLL w OllyDbg, w porównaniu do IDA, mogą pojawiać się różnice w lokalizacji w pamięci.

Zadanie to ma na celu ułatwienie znalezienia poprawnej lokalizacji kodu w OllyDbg w porównaniu do programu IDA. Odpowiedz na poniższe pytania:

1. Które biblioteki DLL są importowane przez Lab06-03.exe (np. PE-bear). Podaj te, które ładują się dynamicznie (IDA: funkcja LoadLibraryA).

Offset	Name	Func. Count	Bound?	OriginalFirstThun	TimeDateStamp
5454	KERNEL32.dll	40	FALSE	54CC	0
5468	NETAPI32.dll	1	FALSE	5570	0
547C	DLL1.dll	1	FALSE	54B8	0
5490	DLL2.dll	2	FALSE	54C0	0

- KERNEL32.dll
- NETAPI32.dll
- DLL1.dll
- DLL2.dll

Dynamicznie **user32.dll**

.text:00403B7F	jnz	short loc_403BC3
.text:00403B81	push	offset aUser32Dll ; "user32.dll"
.text:00403B86	call	ds:LoadLibraryA
.text:00403B8C	mov	edi, eax
.text:00403B8E	cmp	edi, ebx
.text:00403B90	iz	short loc_403BF9

2. Podaj adres bazowy wymagany przez DLL1.dll, DLL2.dll i DLL3.dll (np. w PEvent).

```
.text:10001000
.text:10001000 ; File Name : C:\Users\kali\Desktop\lab5\binaries\DLL1.dll
.text:10001000 ; Format : Portable executable for 80386 (PE)
.text:10001000 ; Imagebase : 10000000
.text:10001000 ; Timestamp : 4E875B33 (Sat Oct 01 18:25:55 2011)
.text:10001000 : Section 1. (virtual address 00001000)
```

Dla wszystkich 3 bibliotek jest to 0x10000000

3. Wykorzystując OllyDbg do debugowania Lab06-03.exe podaj przypisany adres bazowy dla DLL1.dll, DLL2.dll i DLL3.dll.

DLL1.dll

Registers (FPU)	
EAX	00000000
ECX	10001152 DLL1.<ModuleEntryPoint>
EDX	10000000 DLL1.<STRUCT IMAGE_DOS_HEADER>
EBX	00000000
ESP	0019FBD8
EBP	0019FBF4
ESI	0019FBE8
EDI	10001152 DLL1.<ModuleEntryPoint>
EIP	10001152 DLL1.<ModuleEntryPoint>

DLL2.dll

Registers (FPU)	
EAX	00000000
ECX	10001174 DLL2.<ModuleEntryPoint>
EDX	10000000 DLL2.<STRUCT IMAGE_DOS_HEADER>
EBX	00000000
ESP	0019FBD8
EBP	0019FBF4
ESI	0019FBE8
EDI	10001174 DLL2.<ModuleEntryPoint>
EIP	10001174 DLL2.<ModuleEntryPoint>

DLL3.dll

Registers (FPU)	
EAX	00000000
ECX	100011A1 DLL3.<ModuleEntryPoint>
EDX	10000000 DLL3.<STRUCT IMAGE_DOS_HEADER>
EBX	00000000
ESP	0019FBD8
EBP	0019FBF4
ESI	0019FBE8
EDI	100011A1 DLL3.<ModuleEntryPoint>
EIP	100011A1 DLL3.<ModuleEntryPoint>

4. Opisz działanie importowanej funkcji z DLL1.dll wywoływanej przez Lab06-03.exe.

CPU - main thread, module Lab06-03	
00401000	\$ 55 PUSH EBP
00401001	8BEC MOV EBP,ESP
00401003	83EC 1C SUB ESP,1C
00401006	FF15 00504000 CALL DWORD PTR DS:[&DLL1.DLL1Print]
0040100C	FF15 0C504000 CALL DWORD PTR DS:[&DLL2.DLL2Print]
00401012	FF15 08504000 CALL DWORD PTR DS:[&DLL2.DLL2ReturnJ]
00401018	8945 E8 MOV DWORD PTR SS:[LOCAL.6],EAX
0040101B	6A 00 PUSH 0

Lab06-03.00401000(guessed Arg1,Arg2,Arg3)
 DLL2.DLL2Print
 DLL2.DLL2ReturnJ
 r0verlapped = NULL

Jest to funkcja odpowiedzialna za printowanie

Wybierz C:\Users\kali\Desktop\lab5\binaries\Lab06-03.exe

```
DLL 1 mystery data 3768
DLL 2 mystery data 252
DLL 3 mystery data 7844032
```

5. Jaka nazwę pliku wykorzystuje funkcja WriteFile podczas zapisu (Lab06-03.exe w związku z plikiem DLL2.dll)?

```

00008020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00008030  74 65 6D 70 2E 74 78 74 00 00 00 00 44 4C 4C 20 temp.txt...DLL
00008040  32 20 6D 79 73 74 65 72 79 20 64 61 74 61 20 25 2 mystery data %
00008050  64 0A 00 00 EF 1C 00 10 00 B4 00 10 00 00 00 00 d.....
00008060  00 F4 00 10 01 01 00 00 00 00 00 00 00 00 00 00

```

temp.txt

6. Skąd pobierane są dane dla drugiego parametru z funkcji NetScheduleJobAdd?

```

.text:00401018  mov     [ebp+hFile], eax
.text:0040101B  push    0 ; lpOverlapped
.text:0040101D  lea     eax, [ebp+NumberOfBytesWritten]
.text:00401020  push    eax ; lpNumberOfBytesWritten
.text:00401021  push    17h ; nNumberOfBytesToWrite
.text:00401023  push    offset aMalwareanalysisi ; "malwareanalysisbook.com"
.text:00401028  mov     ecx, [ebp+hFile]
.text:0040102B  push    ecx ; hFile
.text:0040102C  call    ds:WriteFile
.text:00401032  mov     edx, [ebp+hFile]
.text:00401035  push    edx ; hObject
.text:00401036  call    ds:CloseHandle
.text:0040103C  push    offset LibFileName ; "DLL3.dll"
.text:00401041  call    ds:LoadLibraryA
.text:00401047  mov     [ebp+hModule], eax
.text:0040104A  push    offset ProcName ; "DLL3Print"
.text:0040104F  mov     eax, [ebp+hModule]
.text:00401052  push    eax ; hModule
.text:00401053  call    ds:GetProcAddress
.text:00401059  mov     [ebp+var_8], eax
.text:0040105C  call    [ebp+var_8]
.text:0040105F  push    offset aDll3getstructu ; "DLL3GetStructure"
.text:00401064  mov     ecx, [ebp+hModule]

```

malwareanalysisbook.com

7. Podczas uruchamiania lub debugowania programu można zobaczyć, że wyświetla on trzy fragmenty tajemniczych danych. Z czym są powiązane: DLL 1 mystery data, DLL 2 mystery data i DLL 3 mystery data?

Nie było śladów wartości w samym pliku Lab06-03.exe

Są powiązane z samymi bibliotekami.

Po odpaleniu bibliotek w programie IDA można zobaczyć, że wartość tych mystery data to wartość PID

```
.text:10001000 ; Attributes: bp-based frame
.text:10001000
.text:10001000 sub_10001000 proc near ; CODE XREF: DllEntryPoint+4B↓p
.text:10001000 push ebp
.text:10001001 mov ebp, esp
.text:10001003 call ds:GetCurrentProcessId
.text:10001009 mov dword_10008030, eax
.text:1000100E mov al, 1
.text:10001010 pop ebp
.text:10001011 retn 0Ch
.text:10001011 sub_10001000 endp
.text:10001011
.....
```

W przypadku DLL2.dll chodzi o nawiązanie do pliku temp.txt

```
.text:10001000
.text:10001000 ; ===== S U B R O U T I N E =====
.text:10001000
.text:10001000 ; Attributes: bp-based frame
.text:10001000
.text:10001000 sub_10001000 proc near ; CODE XREF: DllEntryPoint+4B↓p
.text:10001000 push ebp
.text:10001001 mov ebp, esp
.text:10001003 push 0 ; hTemplateFile
.text:10001005 push 80h ; '€' ; dwFlagsAndAttributes
.text:1000100A push 2 ; dwCreationDisposition
.text:1000100C push 0 ; lpSecurityAttributes
.text:1000100E push 0 ; dwShareMode
.text:10001010 push 40000000h ; dwDesiredAccess
.text:10001015 push offset FileName ; "temp.txt"
.text:1000101A call ds:CreateFileA
.text:10001020 mov dword_10008078, eax
.text:10001025 mov al, 1
.text:10001027 pop ebp
.text:10001028 retn 0Ch
.text:10001028 sub_10001000 endp
.text:10001028
```

W przypadku DLL3.dll wartość jest zależna od funkcji ping do strony www.malwareanalysisbook.com

```
.text:10001000 ; Attributes: bp-based frame
.text:10001000
.text:10001000 sub_10001000 proc near ; CODE XREF: DllEntryPoint+484p
.text:10001000 lpMultiByteStr = dword ptr -4
.text:10001000
.text:10001000 push ebp
.text:10001001 mov ebp, esp
.text:10001003 push ecx
.text:10001004 mov [ebp+lpMultiByteStr], offset aPingWwwMalware ; "ping www.malwareanalysisbook.com"
.text:10001008 push 32h ; '2' ; cchWideChar
.text:1000100D push offset WideCharStr ; lpWideCharStr
.text:10001012 push 0FFFFFFFh ; cbMultiByte
.text:10001014 mov eax, [ebp+lpMultiByteStr]
.text:10001017 push eax ; lpMultiByteStr
.text:10001018 push 0 ; dwFlags
.text:1000101A push 0 ; CodePage
.text:1000101C call ds:MultiByteToWideChar
.text:10001022 mov dword_1000B0AC, offset WideCharStr
.text:1000102C mov dword_1000B0A0, 36EE80h
.text:10001036 mov dword_1000B0A4, 0
.text:10001040 mov byte_1000B0A8, 7Fh
.text:10001047 mov byte_1000B0A9, 11h
.text:1000104E mov al, 1
.text:10001050 mov esp, ebp
.text:10001052 pop ebp
.text:10001053 retn 0Ch
.text:10001053 sub_10001000 endp
```

8. W jaki sposób można załadować DLL2.dll do IDA, aby było to zgodne z adresem ładowania zastosowanym przez OllyDbg?

Należy zmienić bazę programu

