

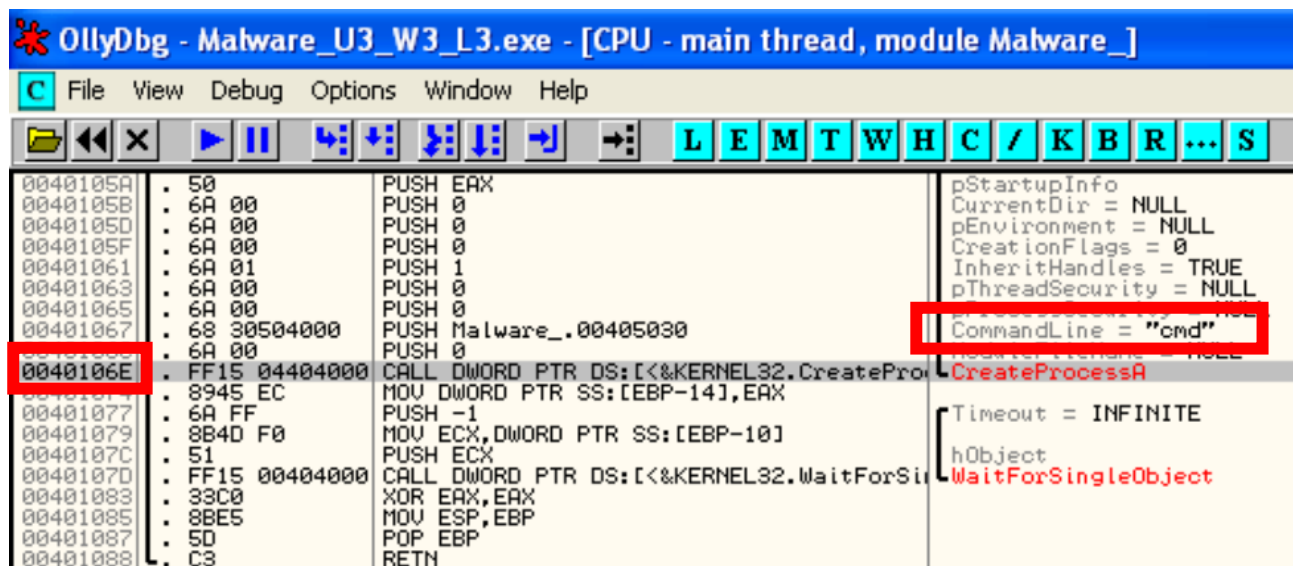
ANALISI DINAMICA AVANZATA

TASK:

1. Qual è il valore del parametro CommandLine che viene passato sullo stack grazie alla chiamata di funzione CreateProcess locata all'indirizzo 0040106E
2. Inserire un breakpoint software all'indirizzo 004015A3. Qual è il valore del registro EDX?
3. Eseguire uno step-into ed indicare il valore del registro EDX dopo averlo fatto
4. Motivare la risposta del task 3
5. Che istruzione è stata eseguita?
6. Inserire un secondo breakpoint all'indirizzo di memoria 004015AF. Qual è il valore del registro ECX?
7. Eseguire uno step-into. Qual è ora il valore di ECX?
8. Spiegare quale istruzione è stata eseguita
9. **BONUS:** Spiegare a grandi linee il funzionamento del malware.

TASK 1: Qual è il valore del parametro CommandLine che viene passato sullo stack grazie alla chiamata di funzione CreateProcess locata all'indirizzo 0040106E

Ci si sposta nell'indirizzo di memoria **0040106E**, dove troviamo il parametro CommandLine come valore **cmd**, prompt dei comandi **Windows**.

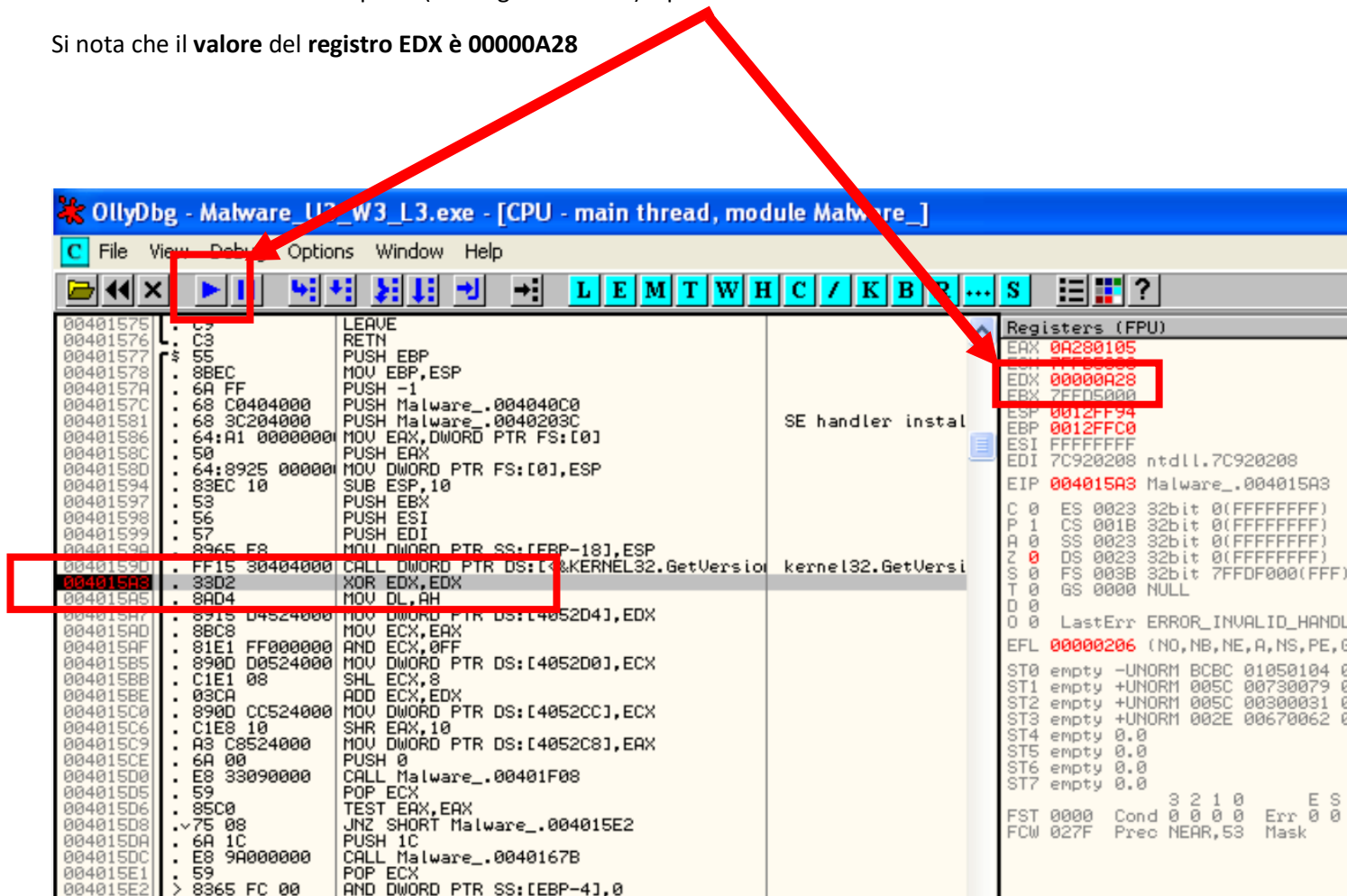


TASK 2: Inserire un breakpoint software all'indirizzo 004015A3. Qual è il valore del registro EDX?

Un software breakpoint permette di fermare il programma quando una data istruzione viene eseguita. Ad esempio, è possibile configurare un software breakpoint su una chiamata di funzione per studiarne i dettagli, oppure all'inizio di un ciclo per capire di cosa si tratta.

Dunque ci sposta all'indirizzo **004015A3** dove con tasto destro del mouse → Breakpoint → Toggle inseriamo un software breakpoint (rettangolo in rosso) e poi **PLAY**.

Si nota che il **valore** del registro **EDX** è **00000A28**



TASK 3-4-5: Eseguire uno STEP-INTO ed indicare il valore del registro EDX dopo averlo fatto, motivare la risposta del task 3. Che istruzione è stata eseguita?

STEP-INTO è una tecnica di debug che ci consente di entrare nel codice della funzione a fronte di una chiamata di funzione, permettendoci di analizzare il suo contenuto e la sua implementazione. È utile quando si desidera analizzare il comportamento di una specifica funzione custom anziché una funzione standard di libreria.

Dopo aver eseguito la tecnica di STEP-INTO, si nota che il valore del registro EDX è pari a **ZERO**, in quanto l'istruzione **XOR EDX, EDX** (dentro la quale siamo entrati) inizializza la variabile EDX a 0.

The screenshot shows the OllyDbg interface with the assembly window and the registers window. Red arrows indicate the steps to follow: one arrow points to the assembly window, and another points to the registers window.

Assembly Window:

Address	Disassembly	Comment
00401575	C9	LEAVE
00401576	C3	RETN
00401577	55	PUSH EBP
00401578	8BEC	MOV EBP,ESP
0040157A	6A FF	PUSH -1
0040157C	68 C0404000	PUSH Malware_.004040C0
00401581	68 3C204000	PUSH Malware_.0040203C
00401586	64:A1 00000000	MOV EAX,DWORD PTR FS:[0]
0040158C	50	PUSH EAX
0040158D	64:8925 00000000	MOV DWORD PTR FS:[0],ESP
00401594	83EC 10	SUB ESP,10
00401597	53	PUSH EBX
00401598	56	PUSH ESI
00401599	57	PUSH EDI
0040159A	8965 E8	MOV DWORD PTR SS:[EBP-18],ESP
0040159D	FF15 30404000	CALL DWORD PTR DS:[<&KERNEL32.GetVersion
004015A3	33D2	XOR EDX,EDX
004015A5	8AD4	MOV ECX,EAX
004015A7	8915 D4524000	MOV DWORD PTR DS:[4052D4],EDX
004015AD	8BC8	MOV ECX,EAX
004015AF	81E1 FF000000	AND ECX,0FF
004015B5	890D D0524000	MOV DWORD PTR DS:[4052D0],ECX
004015B8	C1E1 08	SHL ECX,8
004015BE	03CA	ADD ECX,EDX
004015C0	890D CC524000	MOV DWORD PTR DS:[4052CC],ECX
004015C6	C1E8 10	SHR EAX,10
004015C9	A3 C8524000	MOV DWORD PTR DS:[4052C8],EAX
004015CE	6A 00	PUSH 0
004015D0	E8 33090000	CALL Malware_.00401F08
004015D5	59	POP ECX
004015D6	85C0	TEST EAX,EAX
004015D8	75 08	JNZ SHORT Malware_.004015E2
004015DA	6A 1C	PUSH 1C

Registers (FPU) Window:

Register	Value
EAX	0A280105
EDX	00000000
ECX	0A280105
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C920208 ntdll.7C920208
EIP	004015A5 Malware_.004015A5
C 0	ES 0023 32bit 0(FFFFFFFF)
P 1	CS 001B 32bit 0(FFFFFFFF)
A 0	SS 0023 32bit 0(FFFFFFFF)
Z 1	DS 0023 32bit 0(FFFFFFFF)
S 0	FS 003B 32bit 7FFDF000(FFF)
T 0	GS 0000 NULL
D 0	
O 0	LastErr ERROR_INVALID_HANDLE (00000000)
EFL	00010246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0	empty -UNORM BCBC 01050104 004D005C
ST1	empty +UNORM 005C 00730079 00730069
ST2	empty +UNORM 005C 00300031 00310067
ST3	empty +UNORM 002E 00670062 00640079
ST4	empty 0.0
ST5	empty 0.0
ST6	empty 0.0
ST7	empty 0.0
FST	0000 Cond 0 0 0 0 Err 0 0 0 0 0 0
FCW	027F Prec NEAR,53 Mask 1 1 1 1

TASK 6: Inserire un secondo breakpoint all'indirizzo di memoria 004015AF. Qual è il valore del registro ECX?

Dopo aver inserito un **secondo breakpoint** ed aver cliccato **PLAY**, il valore del registro ECX è pari a 0A280105

The screenshot shows the OllyDbg interface for the file 'Malware_U3_W3_L3.exe'. The CPU window displays assembly instructions. A breakpoint is set at address 004015AF, which is highlighted in red. The instruction at this address is 'AND ECX, 0FF'. The Registers (FPU) window on the right shows the current values of the registers. The ECX register is highlighted in red and contains the value 0A280105. The EIP register also points to 004015AF.

Address	Disassembly	Comment
00401575	C9	LEAVE
00401576	C3	RETN
00401577	55	PUSH EBP
00401578	8BEC	MOV EBP, ESP
00401579	6A FF	PUSH -1
0040157C	68 C0404000	PUSH Malware_.004040C0
00401581	68 3C204000	PUSH Malware_.0040203C
00401586	64:A1 00000000	MOV EAX, DWORD PTR FS:[0]
0040158C	50	PUSH EAX
0040158D	64:8925 00000000	MOV DWORD PTR FS:[0], ESP
00401594	83EC 10	SUB ESP, 10
00401597	53	PUSH EBX
00401598	56	PUSH ESI
00401599	57	PUSH EDI
0040159A	8965 E8	MOV DWORD PTR SS:[EBP-18], ESP
0040159D	FF15 30404000	CALL DWORD PTR DS:[<&KERNEL32.GetVersion
004015A3	33D2	XOR EDX, EDX
004015A5	8AD4	MOV DL, AH
004015A7	8915 04524000	MOV DWORD PTR DS:[4052D4], EDX
004015AD	8BC8	MOV ECX, EAX
004015AF	81E1 FF000000	AND ECX, 0FF
004015B0	8965 E8	MOV DWORD PTR SS:[EBP-18], ESP
004015B8	C1E1 08	SHL ECX, 8
004015BE	03CA	ADD ECX, EDX
004015C0	890D CC524000	MOV DWORD PTR DS:[4052CC], ECX
004015C6	C1E8 10	SHR EAX, 10
004015C9	A3 C8524000	MOV DWORD PTR DS:[4052C8], EAX
004015CE	6A 00	PUSH 0
004015D0	E8 33090000	CALL Malware_.00401F08
004015D5	59	POP ECX
004015D6	85C0	TEST EAX, EAX
004015D8	75 08	JNZ SHORT Malware_.004015E2
004015DA	6A 1C	PUSH 1C
004015DC	E8 9A000000	CALL Malware_.0040167B

Register	Value
EAX	0A280105
ECX	0A280105
EDX	00000000
EBX	7FFD5000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C920208
EIP	004015AF
C 0	ES 0023 32bit 0(FFFFFFFF)
P 1	CS 001B 32bit 0(FFFFFFFF)
A 0	SS 0023 32bit 0(FFFFFFFF)
Z 1	DS 0023 32bit 0(FFFFFFFF)
S 0	FS 003B 32bit 7FFDF000(FFF)
T 0	GS 0000 NULL
D 0	
O 0	LastErr ERROR_INVALID_HANDLE (00000000)
EFL	00000246 (NO, NB, E, BE, NS, PE, GE, LE)
ST0	empty -UNORM BCBC 01050104 004D005C
ST1	empty +UNORM 005C 00730079 00730069
ST2	empty +UNORM 005C 00300031 00310067
ST3	empty +UNORM 002E 00670062 00640079
ST4	empty 0.0
ST5	empty 0.0
ST6	empty 0.0
ST7	empty 0.0
FST	0000 Cond 0 0 0 0 Err 0 0 0 0 0 0
FCW	027F Prec NEAR, 53 Mask 1 1 1 1

TASK 7-8: Eseguire uno STEP-INTO. Qual è ora il valore di ECX? Spiegare quale istruzione è stata eseguita

Eseguiamo nuovamente uno STEP-INTO. Il valore del registro ECX attualmente è **00000005**

In questo caso è stata eseguita l'istruzione **AND ECX, 0FF** che esegue l'AND logico tra il valore del registro ECX il valore 0FF.

OllyDbg - Malware_U3_W3_L3.exe - [CPU - main thread, module Malware_]

File View Debug Options Window Help

Assembly window:

Address	Disassembly	Comment
00401575	LEAVE	
00401576	RETN	
00401577	PUSH EBP	
00401578	MOV EBP,ESP	
0040157A	PUSH -1	
0040157C	PUSH Malware_.004040C0	
00401581	PUSH Malware_.0040203C	
00401586	MOV EAX,DWORD PTR FS:[0]	SE handler instal
0040158C	PUSH EAX	
0040158D	MOV DWORD PTR FS:[0],ESP	
00401594	SUB ESP,10	
00401597	PUSH EBX	
00401598	PUSH ESI	
00401599	PUSH EDI	
0040159A	MOV DWORD PTR SS:[EBP-18],ESP	
0040159D	CALL DWORD PTR DS:[<&KERNEL32.GetVersion	kernel32.GetVersi
004015A3	XOR EDX,EDX	
004015A5	MOV DL,AH	
004015A7	MOV DWORD PTR DS:[4052D4],EDX	
004015AD	MOV ECX,EAX	
004015AF	AND ECX,0FF	
004015B5	MOV DWORD PTR DS:[4052D0],ECX	
004015B8	SHL ECX,8	
004015BE	ADD ECX,EDX	
004015C0	MOV DWORD PTR DS:[4052CC],ECX	
004015C6	SHR EAX,10	
004015C9	MOV DWORD PTR DS:[4052C8],EAX	
004015CE	PUSH 0	
004015D0	CALL Malware_.00401F08	
004015D5	POP ECX	
004015D6	TEST EAX,EAX	
004015D8	JNZ SHORT Malware_.004015E2	
004015DA	PUSH 1C	

Registers (FPU):

Register	Value
EAX	004015B5
ECX	00000005
EDX	00000000
EBX	7FFD5000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C920208 ntdll.7C920208
EIP	004015B5 Malware_.004015B5
C 0	ES 0023 32bit 0(FFFFFFFF)
P 1	CS 001B 32bit 0(FFFFFFFF)
A 0	SS 0023 32bit 0(FFFFFFFF)
Z 0	DS 0023 32bit 0(FFFFFFFF)
S 0	FS 003B 32bit 7FFDF000(FFF)
T 0	GS 0000 NULL
D 0	
O 0	LastErr ERROR_INVALID_HANDLE (00000000)
EFL	00010206 (NO,NB,NE,A,NS,PE,GE,G)
ST0	empty -UNORM BCBC 01050104 00400000
ST1	empty +UNORM 005C 00730079 00730000
ST2	empty +UNORM 005C 00300031 00310000
ST3	empty +UNORM 002E 00670062 00640000
ST4	empty 0.0
ST5	empty 0.0
ST6	empty 0.0
ST7	empty 0.0
FST	0000 Cond 0 0 0 0 Err 0 0 0 0
FCW	027F Prec NEAR,53 Mask 1 1 1

BONUS: Spiegare a grandi linee il funzionamento del malware.

Ricavato l'hash da md5depp e inserito su VirusTotal, si apprende che 43 vendor su 71 hanno identificato il file come **maligno**, nello specifico come un **Trojan**.

```
C:\> Prompt dei comandi
Microsoft Windows XP [Versione 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Epicode_user>cd Desktop
C:\Documents and Settings\Epicode_user\Desktop>cd md5depp-4.3
C:\Documents and Settings\Epicode_user\Desktop\md5depp-4.3>md5depp.exe "C:\Documents and Settings\Epicode_user\Desktop\Malware_U3_W3_L3
C:\Documents and Settings\Epicode_user\Desktop\Malware_U3_W3_L3: No such file or directory
C:\Documents and Settings\Epicode_user\Desktop\md5depp-4.3>md5depp.exe "C:\Documents and Settings\Epicode_user\Desktop\Malware_U3_W3_L3.exe"
251f4d0caf6eadae453488f9c9c0ea95 C:\Documents and Settings\Epicode_user\Desktop\Malware_U3_W3_L3.exe
C:\Documents and Settings\Epicode_user\Desktop\md5depp-4.3>_
```



Search for a hash, domain, IP address, URL or gain additional context and threat landscape visibility with [VT ENTERPRISE](#).

251f4d0caf6eadae453488f9c9c0ea95

43
/ 71

Community Score

43 security vendors and no sandboxes flagged this file as malicious

Reanalyze

Sin

f153dfac09dd69809c3bbf68270a38ee3701f44220c7bf181c14a68c138133

Size: 24.00 KB

Last Analysis Date: 13 days ago

Lab09-02.exe

peexe idle armadillo checks-user-input

DETECTION

DETAILS

RELATIONS

BEHAVIOR

COMMUNITY 9

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to [automate checks](#).

Popular threat label

trojan.genericrxet/neanvzc

Threat categories

trojan

Family labels

genericrxet neanvzc r002c0plk20

Security vendors' analysis

Do you want to

Alibaba	Trojan.Win32/Generic.5a8eecd3	ALYac	Application.Agent.AHB
Antiy-AVL	Trojan/Win32.BTSGeneric	Arcabit	Application.Agent.AHB
Avast	Win32/Malware-gen	AVG	Win32/Malware-gen
BitDefender	Application.Agent.AHB	BitDefenderTheta	Gen.NN.ZexaF.36270.bmW@aaP10K
CrowdStrike Falcon	Win/malicious_confidence_100% (W)	Cybereason	Malicious.caf6ea
Cylance	Unsafe	Cynet	Malicious (score: 100)

VirusTotal ci informa che il file eseguibile importa le librerie:

- **KERNEL32.dll**: libreria piuttosto comune che **contiene le funzioni principali per interagire con il sistema operativo**, ad esempio la **manipolazione dei file**, la **gestione della memoria**
- **WS2_32.dll**: libreria che **contiene le funzioni di network**, come le **socket**, le funzioni **connect**, **bind**.

Header

Target Machine Intel 386 or later processors and compatible processors
Compilation Timestamp 2011-04-30 16:41:06 UTC
Entry Point 5495
Contained Sections 3

Sections

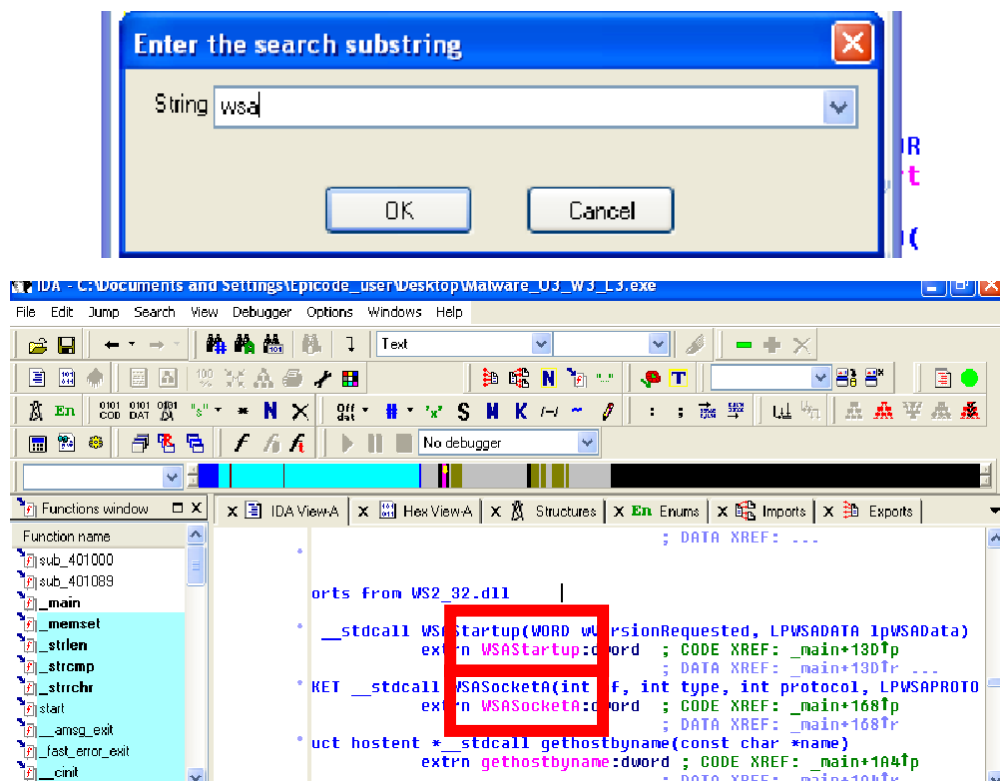
Name	Virtual Address	Virtual Size	Raw Size	Entropy	MD5	Chi2
.text	4096	11366	12288	6.25	6bc8373e9f753cd282af864ad71269c2	131912.97
.rdata	16384	2000	4096	3.17	40e860eea4ba1be6d0d3936b0dbbf076	393924.88
.data	20480	2012	4096	0.69	6103ee24e83b7788dacdba6a1fb83d0	923851.88

Imports

+ KERNEL32.dll
+ WS2_32.dll

Ho utilizzato altresì **IDA Pro**, dove tramite **JUMP BY NAME**, ho rintracciato la funzione **WSAStartup**, usata per allocare risorse che verranno poi utilizzate dalle librerie del networking.

Vi è altresì la funzione **WSASocketA**, la quale potrebbe essere usata per creare un nuovo socket utilizzando la libreria **WS2_32.dll**.



CONCLUSIONI

Si è appurato che il file eseguibile oggetto d'interesse importa la libreria **KERNEL32.dll**, libreria usata per la creazione di un nuovo processo "**cmd**" tramite la funzione "**CreateProcess**".

L'eseguibile oggetto di interesse importa altresì la libreria **WS2_32.dll**, grazie alla quale vengono usate le funzioni **WSAStartup** per allocare risorse che verranno usate per il networking e la funzione **WSASocketA**, la quale viene usata per creare un nuovo socket.

Al netto di questa breve analisi, si può ipotizzare che ci troviamo di fronte ad una probabilissima **backdoor**.