

## Traccia:

Fate riferimento al malware: `Malware_U3_W3_L3`, presente all'interno della cartella `Esercizio_Pratico_U3_W3_L3` sul desktop della macchina virtuale dedicata all'analisi dei malware.

Rispondete ai seguenti quesiti utilizzando **OllyDBG**.

1. All'indirizzo `0040106E` il Malware effettua una chiamata di funzione alla funzione «*CreateProcess*». Qual è il valore del parametro «*CommandLine*» che viene passato sullo stack? (1)
2. Inserite un *breakpoint software* all'indirizzo `004015A3`. Qual è il valore del registro `EDX`? (2) Eseguite a questo punto uno «*step-into*». Indicate qual è ora il valore del registro `EDX` (3) motivando la risposta (4). Che istruzione è stata eseguita? (5)
3. Inserite un secondo breakpoint all'indirizzo di memoria `004015AF`. Qual è il valore del registro `ECX`? (6) Eseguite uno *step-into*. Qual è ora il valore di `ECX`? (7) Spiegate quale istruzione è stata eseguita (8).

**OllyDbg** è un debugger per il software di tipo reverse engineering, comunemente utilizzato per analizzare e modificare programmi binari eseguibili. È popolare tra gli sviluppatori di software, ricercatori di sicurezza e appassionati di hacking per esaminare il funzionamento interno di programmi e identificare vulnerabilità o comportamenti indesiderati.

Avviato il tool sulla nostra macchina virtuale ed aperto il malware da analizzare ci troveremo di fronte questa schermata:

OllyDbg - Malware\_U3\_W3\_L3.exe - [CPU - main thread, module ntdll]

File View Debug Options Window Help

LEMTWHC / KBR...S

77950103 895C24 00 MOV DWORD PTR SS:[ESP+8],EBX  
 77950104 E9 89C0C000 JMP ntdll.77979E8A  
 77950105 8DA424 00000000 LEA ESP,DWORD PTR SS:[ESP]  
 77950106 8DA424 00000000 LEA ESP,DWORD PTR SS:[ESP]  
 77950107 90 NOP  
 77950108 8D04 MOV EDX,ESP  
 77950109 0F34 SYSENTER  
 7795010A C3 RETN  
 7795010B 8DA424 00000000 LEA ESP,DWORD PTR SS:[ESP]  
 7795010C 8D0424 00 LEA EDX,DWORD PTR SS:[ESP]  
 7795010D CD 2E INT 2E  
 7795010E C3 RETN  
 7795010F 90 NOP  
 77950110 ADD BYTE PTR DS:[EAX],AL  
 77950111 ADD BYTE PTR DS:[EAX],AL  
 77950112 SBB BYTE PTR DS:[ECX+4CE7],CL  
 77950113 ADD BYTE PTR DS:[EAX],AL  
 77950114 50 PUSH EAX  
 77950115 51 PUSH ECX  
 77950116 ADD DWORD PTR DS:[EAX],EAX  
 77950117 ADD DWORD PTR DS:[EAX],EAX  
 77950118 ADD BYTE PTR DS:[EAX],AL  
 77950119 F0:07 LOCK POP ES  
 7795011A E8 07000020 CALL 7795021C  
 7795011B ADD AL,BYTE PTR DS:[ECX],AL  
 7795011C ADD AL,AH  
 7795011D AND DWORD PTR DS:[ECX],EAX  
 7795011E AND BYTE PTR DS:[EAX+8C000141],AL  
 7795011F MOV EBX,BC58000A  
 77950120 OR AL,BYTE PTR DS:[EAX]  
 77950121 XOR DWORD PTR DS:[EDI+B915000A],B75D0000  
 77950122 OR AL,BYTE PTR DS:[EAX]  
 77950123 51 PUSH ECX  
 77950124 B5 0A MOV CH,0A  
 77950125 0080 BB0A00BE ADD BYTE PTR SS:[EBP+BE000ABB],CL  
 77950126 73 07 JNB SHORT ntdll.77950246  
 77950127 00F1 ADD CL,DH  
 77950128 AND DWORD PTR DS:[EDX],EAX  
 77950129 ADD BYTE PTR DS:[ECX+21],BL  
 7795012A ADD AL,BYTE PTR DS:[EAX]  
 7795012B F0:26:0300 LOCK ADD EAX,DWORD PTR ES:[EAX]  
 7795012C E9 03 LOOPNE SHORT ntdll.77950221  
 7795012D 07 POP ES  
 7795012E 00F0 ADD AL,DH  
 7795012F D307 ROL DWORD PTR DS:[EDI],CL  
 77950130 9000 ADD AL,DL  
 77950131 0087 ROL DWORD PTR DS:[EDI],CL

Registers (FPU)  
 EAX 00401577 Malware\_...<ModuleEntryPoint>  
 ECX 00000000  
 EDI 00000000  
 EBX 7EFDE000  
 ESP 0018FFF0  
 EBP 00000000  
 CSI 00000000  
 EDI 00000000  
 EIP 779501C8 ntdll.779501C8  
 CS 002B 32bit 0(FFFFFFFF)  
 DS 002B 32bit 0(FFFFFFFF)  
 SS 002B 32bit 0(FFFFFFFF)  
 FS 002B 32bit 0(FFFFFFFF)  
 GS 002B 32bit 0(FFFFFFFF)  
 LastErr ERROR\_SUCCESS (00000000)  
 EFL 00000202 (NO,NB,NE,R,NS,PO,GE,G)  
 ST0 empty 0.0  
 ST1 empty 0.0  
 ST2 empty 0.0  
 ST3 empty 0.0  
 ST4 empty 0.0  
 ST5 empty 0.0  
 ST6 empty 0.0  
 ST7 empty 0.0  
 FST 0000 Cond 0 0 0 Err 0 0 0 0 0 0 (GT)  
 FCW 027F Prec NEAR,53 Mask 1 1 1 1 1

Stack SS:[0018FFF8]=00000000

Address	Hex dump	ASCII
00405000	00 00 00 00 00 00 00 00	.....
00405008	00 00 00 00 F8 27 40 00	....*?@.
00405010	00 00 00 00 00 00 00 00	.....
00405018	00 00 00 00 00 00 00 00	.....
00405020	00 00 00 00 00 00 00 00	.....

0018FFF0 00000000 Malware\_...<ModuleEntryPoint>  
 0018FFF4 00401577  
 0018FFF8 00000000  
 0018FFFC 00000000

Single step event at ntdll.779501C8 - use Shift+F7/F8/F9 to pass execution to program

#### «DisassemblerWindow»:

questa schermata contiene le istruzioni che sono eseguite dalla CPU, con l'aggiunta di alcuni commenti che inserisce OllyDBG direttamente in base al suo DB.

#### «RegisterWindow»:

Essa riporta lo stato dei registri e del loro valore al momento del breakpoint.  
 Questo pannello è fondamentale per capire lo stato attuale del programma. Quando un registro cambia di valore, esso viene evidenziato in rosso in figura.





#### «StackWindow»:

Essa mostra lo stato attuale dello stack in memoria per il programma/funzione che è attualmente in esecuzione.

#### «Memory dumpWindow»:

In questa sezione è incluso il contenuto degli indirizzi di memoria del programma in esecuzione.

Nella tabella di seguito trovate la funzione, una descrizione della funzione e la relativa icona mostrata da OllyDBG.

FUNZIONE	ICONA	DESCRIZIONE
Run / Play		Viene utilizzato per eseguire il programma che è stato precedentemente caricato in OllyDBG
Pausa		Viene utilizzato per stoppare un programma durante la sua esecuzione
Step-into		Utilizzato per esaminare righe di codice e a fronte di una chiamata di funzione accedere alla sua implementazione
Step-over		Utilizzato per esaminare righe di codice singola. A fronte di una chiamata di una funzione permette di saltare l'implementazione della funzione

## 1.-----|

All'indirizzo **0040106E** il Malware effettua una chiamata alla funzione "CreateProcess"; il valore del parametro "CommandLine" è visibile all'indirizzo **00401067**, ovvero **CMD** → *command prompt di Windows*:

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]	
0040105A	. 50	PUSH EAX	
0040105B	. 6A 00	PUSH 0	
0040105D	. 6A 00	PUSH 0	
0040105F	. 6A 00	PUSH 0	
00401061	. 6A 01	PUSH 1	
00401063	. 6A 00	PUSH 0	
00401065	. 6A 00	PUSH 0	
00401067	. 68 30504000	PUSH Malware_.00405030	
0040106C	. 6A 00	PUSH 0	
0040106E	. FF15 04404000	CALL DWORD PTR DS:[&KERNEL32.CreateProcessA]	pProcessInfo pStartupInfo CurrentDir = NULL pEnvironment = NULL CreationFlags = 0 InheritHandles = TRUE pThreadSecurity = NULL pProcessSecurity = NULL CommandLine = "cmd" ModuleFileName = NULL <b>CreateProcessA</b>
00401074	. 8945 EC	MOV DWORD PTR SS:[EBP-14],EAX	
00401077	. 6A FF	PUSH -1	
00401079	. 8B4D F0	MOV ECX,DWORD PTR SS:[EBP-10]	
0040107C	. 51	PUSH ECX	
0040107D	. FF15 00404000	CALL DWORD PTR DS:[&KERNEL32.WaitForSingleObject]	Timeout = INFINITE hObject <b>WaitForSingleObject</b>
00401083	. 33C0	XOR EAX,EAX	
00401085	. 8BE5	MOV ESP,EBP	
00401087	. 5D	POP EBP	
00401088	. C3	RETN	
00401089	. 55	PUSH EBP	


## 2.-----|

Andiamo ora all'indirizzo indicato in esercizio, ovvero **004015A3**, configuriamo un breakpoint ed eseguiamo premendo su play. Come possiamo vedere in figura il programma si fermerà all'istruzione **XOR EDX, EDX**. Prima che l'istruzione venga eseguita il valore del registro è **00401577**:

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 000000	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	kernel32.GetVersion		
004015A3	33D2	XOR EDX,EDX			
004015A5	8A04	MOV DL,AH			
004015A7	8915 D4524000	MOV DWORD PTR DS:[4052D4],EDX			
004015AD	8BC8	MOV ECX,EAX			
004015AF	81E1 FF000000	AND ECX,0FF			
004015B5	89D0 D0524000	MOV DWORD PTR DS:[4052D0],ECX			
004015B8	C1E1 08	SHL ECX,8			
004015BE	03CA	ADD ECX,EDX			
004015C0	89D0 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			
004015D6	59	POP ECX			
004015D8	85C0	TEST EAX,EAX			
004015DB	75 08	JNZ SHORT Malware_.004015E2			
004015DA	6A 1C	PUSH 1C			
004015DC	E8 9A000000	CALL Malware_.0040167B			
004015E1	59	POP ECX			
004015E2	8365 FC 00	AND DWORD PTR SS:[EBP-4],0			
004015E6	E8 72070000	CALL Malware_.00401D5D			
004015E8	FF15 2C404000	CALL DWORD PTR DS:[<&KERNEL32.GetCommand	C:\GetCommandLineA		
004015F1	A3 D8574000	MOV DWORD PTR DS:[4057D8],EAX			
004015F6	E8 30060000	CALL Malware_.00401C2B			

**Registers (FPU)**

EAX	76DF3388	kernel32.BaseThreadInitThunk
ECX	00000000	
EDX	00401577	Malware_.<ModuleEntryPoint>
EBX	7EFD0000	
ESP	0018FF8C	
EBP	0018FF94	
ESI	00000000	
EDI	00000000	
EIP	00401577	Malware_.<ModuleEntryPoint>
C 0	ES 002B 32bit 0(FFFFFFFF)	
P 1	CS 0023 32bit 0(FFFFFFFF)	
Q 0	SS 002B 32bit 0(FFFFFFFF)	
Z 1	DS 002B 32bit 0(FFFFFFFF)	
S 0	FS 0053 32bit 7EFD0000(FFF)	
T 0	GS 002B 32bit 0(FFFFFFFF)	
D 0	LastErr ERROR_SUCCESS (00000000)	
EFL	00000246	(NO,NB,E,BE,NS,PE,GE,LE)
ST0	empty 0.0	
ST1	empty 0.0	
ST2	empty 0.0	
ST3	empty 0.0	
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 0 0 (GT)	
FCW	027F Prec NEAR,S3 Mask 1 1 1 1 1 1	

Ora clicchiamo sul tasto  per eseguire lo *step-into*. Come si può vedere una volta eseguita la funzione *XOR EDX, EDX*, *EDX* il valore di **EDX** sarà **0**, in quanto eseguire lo *step-into* equivale ad inizializzare a zero la funzione:

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 000000	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	kernel32.GetVersion		
004015A3	33D2	XOR EDX,EDX			
004015A5	8A04	MOV DL,AH			
004015A7	8915 D4524000	MOV DWORD PTR DS:[4052D4],EDX			
004015AD	8BC8	MOV ECX,EAX			
004015AF	81E1 FF000000	AND ECX,0FF			
004015B5	89D0 D0524000	MOV DWORD PTR DS:[4052D0],ECX			
004015B8	C1E1 08	SHL ECX,8			
004015BE	03CA	ADD ECX,EDX			
004015C0	89D0 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			
004015D6	59	POP ECX			
004015D8	85C0	TEST EAX,EAX			
004015DB	75 08	JNZ SHORT Malware_.004015E2			
004015DA	6A 1C	PUSH 1C			
004015DC	E8 9A000000	CALL Malware_.0040167B			
004015E1	59	POP ECX			

**Registers (FPU)**

EAX	1DB10106	
ECX	7EFD0000	
EDX	00000000	
EBX	7EFD0000	
ESP	0018FF5C	
EBP	0018FF88	
ESI	00000000	
EDI	00000000	
EIP	004015A5	Malware_.004015A5
C 0	ES 002B 32bit 0(FFFFFFFF)	
P 1	CS 0023 32bit 0(FFFFFFFF)	
Q 0	SS 002B 32bit 0(FFFFFFFF)	
Z 1	DS 002B 32bit 0(FFFFFFFF)	
S 0	FS 0053 32bit 7EFD0000(FFF)	
T 0	GS 002B 32bit 0(FFFFFFFF)	
D 0	LastErr ERROR_SUCCESS (00000000)	
EFL	00000246	(NO,NB,E,BE,NS,PE,GE,LE)
ST0	empty 0.0	
ST1	empty 0.0	
ST2	empty 0.0	
ST3	empty 0.0	
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 0 0 (GT)	
FCW	027F Prec NEAR,S3 Mask 1 1 1 1 1 1	

3.-----|

Facciamo un'altra prova, andiamo all'indirizzo **004015AF** come da istruzione, impostiamo un breakpoint, avviamo con play, notiamo che il valore di **ECX** è **1DB10106**.

In seguito allo step-into [  ] il valore di **ECX** cambia in **00000006** in quanto viene eseguita l'istruzione **AND ECX, FF:**

00401577	55	PUSH EBP		Registers (FPU)
00401578	8BEC	MOV EBP,ESP		EAX 1DB10106
0040157A	6A FF	PUSH -1		<b>ECX 1DB10106</b>
0040157C	68 C0404000	PUSH Malware_.004040C0		EDX 00000001
00401581	68 3C204000	PUSH Malware_.0040203C	SE handler installation	EBX 7EFD0000
00401586	64:R1 00000000	MOV EAX,DWORD PTR FS:[0]		ESP 0018FF5C
0040158C	50	PUSH EAX		EBP 0018FF88
0040158D	64:8925 000000	MOV DWORD PTR FS:[0],ESP		ESI 00000000
00401594	83EC 10	SUB ESP,10		EDI 00000000
00401597	53	PUSH EBX		EIP 004015AF Malware_.004015AF
00401598	56	PUSH ESI		C 0 ES 002B 32bit 0(FFFFFFFF)
00401599	57	PUSH EDI		P 1 CS 0023 32bit 0(FFFFFFFF)
0040159A	8965 E8	MOV DWORD PTR SS:[EBP-18],ESP		A 0 SS 002B 32bit 0(FFFFFFFF)
0040159B	FF15 30404000	CALL DWORD PTR DS:[<&KERNEL32.GetVersion	kernel32.GetVersion	Z 1 DS 002B 32bit 0(FFFFFFFF)
004015A5	33D2	XOR EDX,EDX		S 0 FS 0053 32bit 7EFD0000(FFF)
004015A6	8AD4	MOV DL,AH		T 0 GS 002B 32bit 0(FFFFFFFF)
004015A7	8915 D4524000	MOV DWORD PTR DS:[4052D4],EDX		D 0
004015A8	8BC8	MOV ECX,EAX		0 0 LastErr ERROR_SUCCESS (00000000)
004015A9	81E1 FF000000	AND ECX,0FF		EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
004015B5	890D D0524000	MOV DWORD PTR DS:[4052D0],ECX		
004015B8	C1E1 08	SHL ECX,8		
004015BE	03CA	ADD ECX,EDX		

00401577	55	PUSH EBP		Registers (FPU)
00401578	8BEC	MOV EBP,ESP		EAX 1DB10106
0040157A	6A FF	PUSH -1		<b>ECX 00000006</b>
0040157C	68 C0404000	PUSH Malware_.004040C0		EDX 00000001
00401581	68 3C204000	PUSH Malware_.0040203C	SE handler installation	EBX 7EFD0000
00401586	64:R1 00000000	MOV EAX,DWORD PTR FS:[0]		ESP 0018FF5C
0040158C	50	PUSH EAX		EBP 0018FF88
0040158D	64:8925 000000	MOV DWORD PTR FS:[0],ESP		ESI 00000000
00401594	83EC 10	SUB ESP,10		EDI 00000000
00401597	53	PUSH EBX		EIP 004015B5 Malware_.004015B5
00401598	56	PUSH ESI		C 0 ES 002B 32bit 0(FFFFFFFF)
00401599	57	PUSH EDI		P 1 CS 0023 32bit 0(FFFFFFFF)
0040159A	8965 E8	MOV DWORD PTR SS:[EBP-18],ESP		A 0 SS 002B 32bit 0(FFFFFFFF)
0040159B	FF15 30404000	CALL DWORD PTR DS:[<&KERNEL32.GetVersion	kernel32.GetVersion	Z 0 DS 002B 32bit 0(FFFFFFFF)
004015A5	33D2	XOR EDX,EDX		S 0 FS 0053 32bit 7EFD0000(FFF)
004015A6	8AD4	MOV DL,AH		T 0 GS 002B 32bit 0(FFFFFFFF)
004015A7	8915 D4524000	MOV DWORD PTR DS:[4052D4],EDX		D 0
004015A8	8BC8	MOV ECX,EAX		0 0 LastErr ERROR_SUCCESS (00000000)
004015A9	81E1 FF000000	AND ECX,0FF		EFL 00000206 (NO,NB,NE,A,NS,PE,GE,G)
004015B5	890D D0524000	MOV DWORD PTR DS:[4052D0],ECX		
004015B8	C1E1 08	SHL ECX,8		
004015BE	03CA	ADD ECX,EDX		