

**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.

- 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)**
- 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)**
- Inserite un secondo breakpoint all'indirizzo di memoria 004015AF. Qual è il valore del registro ECX? **(6)**  
Eseguite un step-into. Qual è ora il valore di ECX? **(7)** Spiegate quale istruzione è stata eseguita **(8)**.
- **BONUS:** spiegare a grandi linee il funzionamento del malware

-1

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 Malware_00405030	CommandLine = "cmd"
0040106C	6A 00	PUSH 0	ModuleFileName = NULL
0040106E	FF15 04404000	CALL DWORD PTR DS:[&KERNEL32.CreateProcessA]	CreateProcessA

Come si vede dall'immagine, il valore che viene passato nella riga **00401067** dal comando **PUSH** e che corrisponde al parametro **CommandLine** della funzione **CreateProcessA** e' **Malware .00405030**

**-2, 3, 4, 5**

Address	Disassembly	Comment	Registers (FPU)
00401598	PUSH ESI		
00401599	PUSH EDI		
0040159A	MOV DWORD PTR SS:[EBP-18], ESP		
0040159B	CALL DWORD PTR DS:[4*kernel32.GetVersion]	kernel32.GetVersion	
0040159C	XOR EDI, EDI		
0040159D	MOV DL, AL		
0040159E	MOV DWORD PTR DS:[405204], EDI		
0040159F	MOV ECX, EAX		
004015A0	ESP_0018FF8C		
004015A1	ESP_0018FF94		
004015A2	ESP_0018FF98		

Come si vede dall'immagine, il valore che si trova nel registro **EDX** prima di avviare il programma dall'entry point e' proprio la riga di entry point

Address	Disassembly	Comment	Registers (RPU)
00401599	PUSH EDI		EAX 10B10105
0040159A	MOV DWORD PTR SS:[EBP-18], ESP		
0040159B	CALL DWORD PTR DS:[<kernel32.GetVersion>]	kernel32.GetVersion	EDX 000010B1
0040159C	XOR EDX, EDX		ESP 0018FF5C
0040159D	MOV DL, AH		ESP 0018FF5B
0040159E	MOV DWORD PTR DS:[4052D4], EDX		ESI 00000000
0040159F	MOV ECX, EAX		
004015A0	CALL FFFFFFFF		

Dopo aver creato il **break point** alla riga **004015A3** ed aver avviato il malware, il valore del registro **EDX** cambia in **00001DB1**

<pre> 00401577      ; PUSH ESI 00401584      MOV DWORD PTR SS:[EBP-18],ESP 00401590      CALL DWORD PTR DS:[&lt;&lt;kernel32.GetVersion&gt;] 00401596      XOR EDX,EDX 004015A2      MOV DL,AH 004015A7      MOV DWORD PTR DS:[4052D41],EDX 004015AD      MOV ECK,EAX 004015AF      AND ECK,EFF 004015B5      MOV DWORD PTR DS:[4052D01],ECK </pre>	<pre> kernel32.GetVersion </pre>	<div style="border: 1px solid black; padding: 2px;"> <div style="background-color: #f0f0f0; padding: 2px;">EAX 00101016</div> <div style="padding: 2px;">ESI 7F525080</div> <div style="border: 2px solid red; padding: 2px;"><div style="background-color: #f0f0f0; padding: 2px;">EDX 00000000</div></div> <div style="padding: 2px;">ESP 0018FFC7</div> <div style="padding: 2px;">EBP 0018FF88</div> <div style="padding: 2px;">ESI 00000000</div> <div style="padding: 2px;">EDI 00000000</div> </div>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Premendo il pulsante **step into**, il programma esegue solamente la riga successiva a dove si trova nel momento di pausa e nell'immagine si puo' vedere il risultato del comando della riga **004015A3**; il registro **EDX** ha cambiato valore in zero; l'operatore logico **XOR** produce in uscita il valore 1 se e solo se i valori in ingresso sono diversi tra loro; mettere 2 valori uguali all'ingresso di **XOR** e' un espediente per resettare il registro a zero in modo veloce.

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

-6, 7, 8

004015A7	8915 D4524000	MOV DWORD PTR DS:[4052D41],ECX	
004015A8	8BC8	MOV ECX, EAX	
004015AF	81E1 FF000000	AND ECX, 0FF	ECX: 1DB10106
004015B5	8900 D0524000	MOV DWORD PTR DS:[4052D01],ECX	EBX: 7EFDE000
004015B8	C1E1 08	SHL ECX, 8	ESP: 0018FF5C

Il valore del registro **ECX** prima dell'esecuzione della riga **004015AF** e' **1DB10106**

004015A7	8915 D4524000	MOV DWORD PTR DS:[4052D41],ECX	
004015A8	8BC8	MOV ECX, EAX	
004015AF	81E1 FF000000	AND ECX, 0FF	ECX: 00000006
004015B5	8900 D0524000	MOV DWORD PTR DS:[4052D01],ECX	EBX: 7EFDE000
004015B8	C1E1 08	SHL ECX, 8	ESP: 0018FF5C
004015BA	03CA	ADD ECX, EDI	EBP: 0018FF88
004015C0	8900 C0524000	MOV DWORD PTR DS:[4052C01],ECX	EI1: 00000000

Usando l'opzione **step into** si vede che e' cambiato in **00000006**; questo perche' l'operatore logico **AND** consegna in uscita un valore 1 solo quando entrambi i valori in ingresso sono 1; convertendo i valori della destinazione e della sorgente in binario e inserendoli nell'operatore logico si ottiene il valore sopra citato.

	ESADECIMALE	BINARIO
<b>Destinazione:</b>	<b>1DB10106 (ECX)</b>	<b>0 0 0 1 1 1 0 1 1 0 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0</b>
<b>Sorgente:</b>	<b>FF</b>	<b>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1</b>
<b>AND</b>		
<b>Risultato:</b>	<b>6</b>	<b>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0</b>

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1