

ESERCIZIO W24D1

L'esercizio seguente espone la traccia sottostante:

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).

3

Primo quesito

CPU - main thread, module Malware_		
00401040	. 8945 E8	MOV DWORD PTR SS:[EBP-18],EAX
00401040	. 8B40 E8	MOV ECX, DWORD PTR SS:[EBP-18]
00401050	. 8940 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

nome valore

pStartupInfo

CurrentDir = NULL

pEnvironment = NULL

CreationFlags = 0

InheritHandles = TRUE

pThreadSecurity = NULL

pProcessSecurity = NULL

CommandLine = "cmd"

ModuleFileName = NULL

CreateProcessA

Osservando l'immagine, notiamo che il valore passato su command line corrisponde a "cmd"

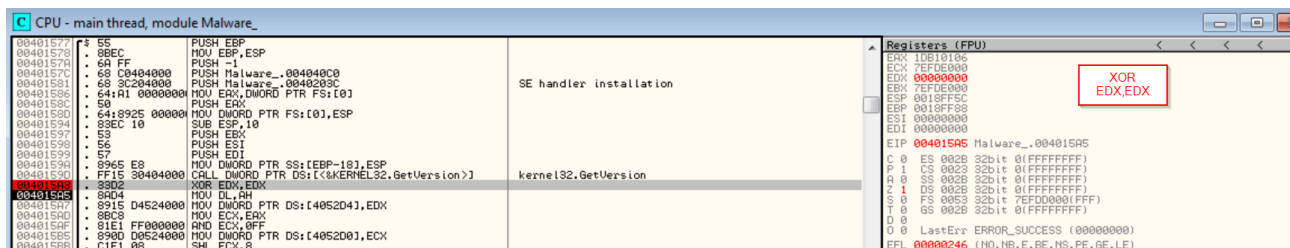
Secondo quesito



Registers (FPU)

Register	Value
EAX	10B10106
ECX	7EFDE000
EDX	00001DB1
EBX	7EFDE000
ESP	0018FF5C
EBP	0018FF58
ESI	00000000
EDI	00000000
EIP	004015A3 Malware_.004015A3

Inseriamo un breakpoint software all'interno dell'indirizzo 004015A3, il valore di EDX è 00001DB1.



Registers (FPU)

Register	Value
EAX	10B10106
ECX	7EFDE000
EDX	00000000
EBX	7EFDE000
ESP	0018FF5C
EBP	0018FF58
ESI	00000000
EDI	00000000
EIP	004015A3 Malware_.004015A3

Una volta eseguita l'istruzione, EDX sfoggia un valore esadecimale corrispondente a 8 zeri. XOR EDX, EDX da come risultato False per tutti i bits, visto che XOR da vero solo se le rotte binarie sono diverse.

Terzo quesito

CPU - main thread, module Malware_

00401577	55	PUSH EBP	
00401578	5B	MOV EBP, ESP	
00401579	6A	PUSH -1	
0040157C	68	PUSH Malware_.004040C0	
00401581	68	PUSH Malware_.004040C0	
00401586	64	MOV ECX, DWORD PTR FS:[0]	SE handler installation
0040158C	5B	PUSH EBX	
0040158D	64	MOV DWORD PTR FS:[0], ESP	
00401594	83	SUB ESP, 10	
00401597	53	PUSH EBX	
00401598	56	PUSH ESI	
00401599	57	PUSH EDI	
0040159A	95	MOV DWORD PTR SS:[EBP-10], ESP	
0040159B	FF	CALL DWORD PTR DS:[<<KERNEL32.GetVersion>]	kernel32.GetVersion
0040159C	33	XOR EDX, EDX	
0040159D	9A	MOV DL, AH	
0040159E	8B	MOV DWORD PTR DS:[4052D4], EDX	
0040159F	8B	MOV ECX, EDX	
004015A0	8B	MOV ECX, EDX	
004015A1	8B	MOV ECX, EDX	
004015A2	8B	MOV ECX, EDX	
004015A3	8B	MOV ECX, EDX	
004015A4	8B	MOV ECX, EDX	
004015A5	8B	MOV ECX, EDX	
004015A6	8B	MOV ECX, EDX	
004015A7	8B	MOV ECX, EDX	
004015A8	8B	MOV ECX, EDX	
004015A9	8B	MOV ECX, EDX	
004015AA	8B	MOV ECX, EDX	
004015AB	8B	MOV ECX, EDX	
004015AC	8B	MOV ECX, EDX	
004015AD	8B	MOV ECX, EDX	
004015AE	8B	MOV ECX, EDX	
004015AF	8B	MOV ECX, EDX	
004015B0	8B	MOV ECX, EDX	
004015B1	8B	MOV ECX, EDX	
004015B2	8B	MOV ECX, EDX	
004015B3	8B	MOV ECX, EDX	
004015B4	8B	MOV ECX, EDX	
004015B5	8B	MOV ECX, EDX	
004015B6	8B	MOV ECX, EDX	
004015B7	8B	MOV ECX, EDX	
004015B8	8B	MOV ECX, EDX	
004015B9	8B	MOV ECX, EDX	
004015BA	8B	MOV ECX, EDX	
004015BB	8B	MOV ECX, EDX	
004015BC	8B	MOV ECX, EDX	
004015BD	8B	MOV ECX, EDX	
004015BE	8B	MOV ECX, EDX	
004015BF	8B	MOV ECX, EDX	
004015C0	8B	MOV ECX, EDX	
004015C1	8B	MOV ECX, EDX	
004015C2	8B	MOV ECX, EDX	
004015C3	8B	MOV ECX, EDX	
004015C4	8B	MOV ECX, EDX	
004015C5	8B	MOV ECX, EDX	
004015C6	8B	MOV ECX, EDX	
004015C7	8B	MOV ECX, EDX	
004015C8	8B	MOV ECX, EDX	
004015C9	8B	MOV ECX, EDX	
004015CA	8B	MOV ECX, EDX	
004015CB	8B	MOV ECX, EDX	
004015CC	8B	MOV ECX, EDX	
004015CD	8B	MOV ECX, EDX	
004015CE	8B	MOV ECX, EDX	
004015CF	8B	MOV ECX, EDX	
004015D0	8B	MOV ECX, EDX	
004015D1	8B	MOV ECX, EDX	
004015D2	8B	MOV ECX, EDX	
004015D3	8B	MOV ECX, EDX	
004015D4	8B	MOV ECX, EDX	
004015D5	8B	MOV ECX, EDX	
004015D6	8B	MOV ECX, EDX	
004015D7	8B	MOV ECX, EDX	
004015D8	8B	MOV ECX, EDX	
004015D9	8B	MOV ECX, EDX	
004015DA	8B	MOV ECX, EDX	
004015DB	8B	MOV ECX, EDX	
004015DC	8B	MOV ECX, EDX	
004015DD	8B	MOV ECX, EDX	
004015DE	8B	MOV ECX, EDX	
004015DF	8B	MOV ECX, EDX	
004015E0	8B	MOV ECX, EDX	
004015E1	8B	MOV ECX, EDX	
004015E2	8B	MOV ECX, EDX	
004015E3	8B	MOV ECX, EDX	
004015E4	8B	MOV ECX, EDX	
004015E5	8B	MOV ECX, EDX	
004015E6	8B	MOV ECX, EDX	
004015E7	8B	MOV ECX, EDX	
004015E8	8B	MOV ECX, EDX	
004015E9	8B	MOV ECX, EDX	
004015EA	8B	MOV ECX, EDX	
004015EB	8B	MOV ECX, EDX	
004015EC	8B	MOV ECX, EDX	
004015ED	8B	MOV ECX, EDX	
004015EE	8B	MOV ECX, EDX	
004015EF	8B	MOV ECX, EDX	
004015F0	8B	MOV ECX, EDX	
004015F1	8B	MOV ECX, EDX	
004015F2	8B	MOV ECX, EDX	
004015F3	8B	MOV ECX, EDX	
004015F4	8B	MOV ECX, EDX	
004015F5	8B	MOV ECX, EDX	
004015F6	8B	MOV ECX, EDX	
004015F7	8B	MOV ECX, EDX	
004015F8	8B	MOV ECX, EDX	
004015F9	8B	MOV ECX, EDX	
004015FA	8B	MOV ECX, EDX	
004015FB	8B	MOV ECX, EDX	
004015FC	8B	MOV ECX, EDX	
004015FD	8B	MOV ECX, EDX	
004015FE	8B	MOV ECX, EDX	
004015FF	8B	MOV ECX, EDX	

Registers (FPU)

EAX	1DB10106
ECX	1DB10106
EDX	00000001
EBX	7EFDE000
ESP	0018FF5C
EBP	0018FF88
ESI	00000000
EDI	00000000
EIP	004015AF Malware_.004015AF
C	0 ES 002B 32bit 0(FFFFFFFF)
P	1 CS 0023 32bit 0(FFFFFFFF)
A	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)
O	0
D	0
LastErr	ERROR_SUCCESS (00000000)
INI	NR F BF NS PF GE IF

Inseriamo un altro breakpoint sull'indirizzo di memoria 004015AF. Il valore del registro ECX è 1DB10106.

Registers (FPU)

EAX	1DB10106
ECX	00000006
EDX	00000001
EBX	7EFDE000
ESP	0018FF5C
EBP	0018FF88
ESI	00000000
EDI	00000000

AND ECX, 0FF

Dopo lo step into, viene effettuata l'istruzione AND tra le rotte binarie di ECX, 0FF per mezzo dell'operatore logico AND. AND ci da vero solo se entrambi i valori risultano veri.

ECX → 11101101100010000000100000110

0FF → 0000000000000000000000001111111

AND → 0000000000000000000000000000110

Esattamente come dice il risultato dei registri