



# Compito S11-L3



# 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

Il valore del parametro è “CMD” ovvero il command prompt di Windows, come si nota nella figura sottostante all'indirizzo 00401067.

00401057	. 8D45 A8	LEA EAX,DWORD PTR SS:[EBP-58]	pStartupInfo
0040105A	. 50	PUSH EAX	CurrentDir = NULL
0040105B	. 6A 00	PUSH 0	pEnvironment = NULL
0040105D	. 6A 00	PUSH 0	CreationFlags = 0
0040105F	. 6A 00	PUSH 0	InheritHandles = TRUE
00401061	. 6A 01	PUSH 1	pThreadSecurity = NULL
00401063	. 6A 00	PUSH 0	pProcessSecurity = NULL
00401065	. 6A 00	PUSH 0	CommandLine = "cmd"
00401067	. 68 30504000	PUSH Malware_.00405030	ModuleFileName = NULL
0040106C	. 6A 00	PUSH 0	CreateProcessA
0040106E	. FF15 04404000	CALL DWORD PTR DS:[<&KERNEL32.CreatePro	
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	hObject
0040107D	. FF15 00404000	CALL DWORD PTR DS:[<&KERNEL32.WaitForSi	WaitForSingleObject
00401083	. 33C0	XOR EAX,EAX	
00401085	. 8BES	MOV ESP,EBP	
00401087	. 5D	POP EBP	

## 2, 3, 4, 5

Una volta configurato il breakpoint, clicchiamo su “play”, il programma si fermerà all'istruzione XOR EDX,EDX. Prima che l'istruzione venga eseguita il valore del registro è “00000A28”. Dopo lo step-into, viene eseguita l'istruzione XOR EDX,EDX che di fatto equivale ad inizializzare a zero una variabile. Quindi, dopo lo step-into il valore di EDX sarà 0.

### Prima

00401590	FF15 30404000	CALL DWORD PTR DS:[<&KERNEL32.GetVersion	kernel32.GetVersion
00401593	33D2	XOR EDX,EDX	
004015A5	8AD4	MOV DL,AH	

### Dopo

00401590	FF15 30404000	CALL DWORD PTR DS:[<&KERNEL32.GetVersion	kernel32.GetVersion
00401593	33D2	XOR EDX,EDX	
004015A5	8AD4	MOV DL,AH	
004015A7	8915 04524000	MOV DWORD PTR DS:[4052041.FDX	

[illegible][illegible]



## 8

Nel dettaglio, l'istruzione esegue l'AND logico sui bit di EAX e del valore esadecimale FF. Per prima cosa portiamo entrambi i valori in formato binario e poi eseguiamo l'AND logico tra i bit.

Esadecimale	Binario
0A280105	0000 1010 0010 1000 0000 0001 0000 0101
FF	0000 0000 0000 0000 0000 0000 1111 1111

Il risultato è 0000 0000 0000 0000 0000 0000 0000 0101  
Che in esadecimale è 5. (oppure 00000005)