

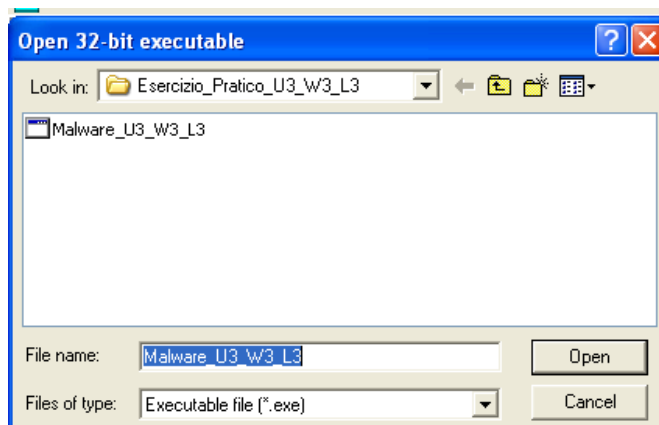
Esercizio 05-04

Traccia: facendo 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:

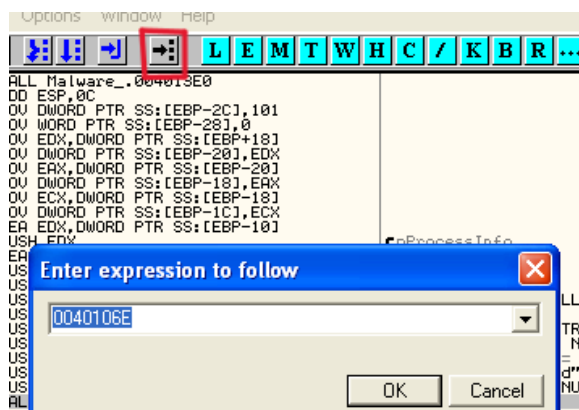
1. All'indirizzo 0040106E il malware effettua una chiamata di funzione alla funzione "CreateProcess". Qual è il valore del parametro "CommandLine" che viene passato sullo stack?
2. Inserite un breakpoint software all'indirizzo 004015A3. Qual è il valore del registro EDX? eseguite a questi punto uno "step-into". Indicate qual è ora il valore del registro EDX motivando la risposta. Che istruzione è stata eseguita?
3. Inserite un secondo breakpoint all'indirizzo di memoria 004015AF. Qual è il valore del registro ECX? Eseguite un step-into. Qual è ora il valore di ECX? Spiegate quale istruzione è stata eseguita.
4. BONUS: ipotizzare il funzionamento del malware

1.

Dopo aver avviato OllyDBG carichiamo l'eseguibile da analizzare



Conoscendo l'indirizzo della funzione che vogliamo analizzare (0040106E) utilizzando l'iconcina apposita possiamo spostarci direttamente all'indirizzo che mettiamo



Eccoci all'indirizzo che abbiamo specificato, adesso andiamo a vedere quale valore viene passato al parametro CommandLine della funzione CreateProcess

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	pProcessInfo
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	. FF15 04404000	CALL DWORD PTR DS:[<&KERNEL32.CreatePro	CreateProcessA
00401074	. 8945 EC	MOV DWORD PTR SS:[EBP-14],EAX	

Il parametro che viene passato a CommandLine è "cmd", il prompt comandi di Windows

0040105A	. 50	PUSH EAX	pStartupInfo
0040105B	. 6A 00	PUSH 0	CurrentDir = NULL
0040105D	. 6A 00	PUSH 0	pEnvironment = NULL
0040105F	. 6A 00	PUSH 0	CreationFlags = 0
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	. FF15 04404000	CALL DWORD PTR DS:[<&KERNEL32.CreatePro	ModuleFileName = NULL
00401074	. 8945 EC	MOV DWORD PTR SS:[EBP-14],EAX	CreateProcessA
00401077	. 6A FF	PUSH -1	Timeout = INFINITE
00401079	. 8B4D F0	MOV ECX,DWORD PTR SS:[EBP-10]	hObject
0040107C	. 51	PUSH ECX	WaitForSingleObject
0040107D	. FF15 00404000	CALL DWORD PTR DS:[<&KERNEL32.WaitForSi	
00401083	. 33C0	XOR EAX,EAX	

2.

Aggiunta di un software breakpoint all'indirizzo 004015A3, in questo punto il registro EDX diventa "0", essendo risultato di uno XOR che compara due registri uguali

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	. 8AD4	MOV DL,AH	
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	
004015BB	. C1E1 08	SHL ECX,8	
004015BE	. 03CA	ADD ECX,EDX	
004015C0	. 890D CC524000	MOV DWORD PTR DS:[4052CC],ECX	
004015C6	. C1F8 10	SHR EAX,10	

Eseguendo il programma con il comando “run” il flusso si bloccherà al breakpoint appena messo

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	. 8AD4	MOV DL,AH	
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	

Il valore di EDX in esadecimale e di A28 (2 600 decimale)

Registers (FPU)	
EAX	0A280105
ECX	7FFD9000
EDX	00000A28
EBX	7FFD9000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	FFFFFFFF

Adesso eseguiamo uno “step - into” con l’icona apposita o F7



Il codice è proseguito all’istruzione successiva

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	. 8AD4	MOV DL,AH	
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	

Il valore di EDX è cambiato, come conseguenza dello XOR tra due valori uguali restituisce “0”

Registers (FPU)	
EAX	0A280105
ECX	7FFD9000
EDX	00000000
EBX	7FFD9000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C910208 ntdll.

3.

Aggiunta di un breakpoint all'indirizzo 004015AF

0040159D	. FF15 30404000	CALL DWORD PTR DS:[<&KERNEL32.GetVersion
004015A3	. 33D2	XOR EDX,EDX
004015A5	. 8AD4	MOV DL,AH
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
004015BB	. C1E1 08	SHL ECX,8
004015BE	. 03CA	ADD ECX,EDX

Cliccando su “run” il flusso del programma prosegue e raggiunge il breakpoint

0040159D	. FF15 30404000	CALL DWORD PTR DS:[<&KERNEL32.GetVersion	kernel32.GetVersion
004015A3	. 33D2	XOR EDX,EDX	
004015A5	. 8AD4	MOV DL,AH	
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	
004015BB	. C1E1 08	SHL ECX,8	
004015BE	. 03CA	ADD ECX,EDX	
004015C0	. 890D CC524000	MOV DWORD PTR DS:[4052CC],ECX	

Qui il valore di ECX è A280105 che in decimale corrisponde a 170 393 861

EAX	0A280105
ECX	0A280105
EDX	00000001
EBX	7FFD9000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C910208

Eseguendo uno “step - into” il codice prosegue all'istruzione successiva

004015A5	. 8AD4	MOV DL,AH
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
004015BB	. C1E1 08	SHL ECX,8
004015BE	. 03CA	ADD ECX,EDX
004015C0	. 890D CC524000	MOV DWORD PTR DS:[4052CC],ECX

Il valore di ECX è cambiato, come risultato dell' AND precedente restituisce il valore 5
(corrispondente a 5 anche in decimale)

Registers (FPU)	
EAX	0A280105
ECX	00000005
EDX	00000001
EBX	7FFD9000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C910208

Risoluzione dell'operazione con i valori in codice binario

FF = 0000 0000 0000 0000 0000 0000 1111 1111

A280105 = 0000 1010 0010 1000 0000 0001 0000 0101

Risultato = 0000 0000 0000 0000 0000 0000 0000 0101 = 5