

# S11/L3

## TRACCIA

Analizzare il malware **Malware\_U3\_W3\_L3** rispondere 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 questo 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: spiegare a grandi linee il funzionamento del malware

## ESERCIZIO

### 1.

Analizzando l'indirizzo di memoria **0040106E** possiamo notare che il valore della funzione è **cmd** come evidenziato in figura.

CMD è il Prompt dei Comandi disponibile nella maggior parte dei sistemi operativi Windows, che viene utilizzato per eseguire dei comandi immessi dall'utente.

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	. 6A 00	PUSH 0	ModuleFileName = NULL
0040106E	. FF15 04404000	CALL DWORD PTR DS:[<&KERNEL32.CreateProcessA>]	CreateProcessA
00401074	. 8945 EC	MOV DWORD PTR SS:[EBP-14],EAX	

### 2.

Andando ad inserire un breakpoint software attraverso il comando "Toggle Breakpoint" sull'indirizzo di memoria **004015A3** possiamo fermare il programma in esecuzione per analizzare più nello specifico il malware senza che il debugger continui a leggerlo. Una volta fatto ciò, possiamo vedere che il valore di EDX corrisponde a **00000A28**

004015A3	. 33D2	XOR EDX,EDX	Registers (FPU)
004015A5	. 8AD4	MOV DL,AH	EAX 0A280105
004015A7	. 8915 D4524000	MOV DWORD PTR DS:[4052D4],EDX	ECX 7FFDC000
004015AD	. 8BC8	MOV ECX,EAX	EDX 00000A28
004015AF	. 81E1 FF000000	AND ECX,0FF	EBX 7FFDC000
004015B5	. 890D D0524000	MOV DWORD PTR DS:[4052D0],ECX	ESP 0012FF94
004015BB	. C1E1 08	SHL ECX,8	EBP 0012FFC0
004015BE	. 03CA	ADD ECX,EDX	ESI FFFFFFFF
004015C0	. 890D CC524000	MOV DWORD PTR DS:[4052CC],ECX	EDI 7C910208 ntdll.7C910208

Con la funzione "step-into" andremo ad analizzare la funzione dove viene implementata. Nel nostro caso possiamo notare che il registro EDX è inizializzato a zero come riportato dallo XOR individuato nella chiamata di funzione.

004015A3	. 33D2	XOR EDX,EDX	Registers (FPU)
004015A5	. 8AD4	MOV DL,AH	EAX 0A280105
004015A7	. 8915 D4524000	MOV DWORD PTR DS:[4052D4],EDX	ECX 7FFDC000
004015AD	. 8BC8	MOV ECX,EAX	EDX 00000000
004015AF	. 81E1 FF000000	AND ECX,0FF	EBX 7FFDC000
004015B5	. 890D D0524000	MOV DWORD PTR DS:[4052D0],ECX	ESP 0012FF94
004015BB	. C1E1 08	SHL ECX,8	EBP 0012FFC0
004015BE	. 03CA	ADD ECX,EDX	ESI FFFFFFFF
004015C0	. 890D CC524000	MOV DWORD PTR DS:[4052CC],ECX	EDI 7C910208 ntdll.7C910208
004015C6	. C1E8 10	SHR EAX,10	EIP 004015A5 Malware_.004015A5
004015C9	. A3 C8524000	MOV DWORD PTR DS:[4052C8],EAX	

3.

Ripetendo lo stesso procedimento come nel punto **2** inseriremo un breakpoint software sull'indirizzo di memoria **004015AF**. Possiamo vedere come in figura che il valore di ECX corrisponde a **0A280105**.

00401575	. C9	LEAVE
00401576	. C3	RETN
00401577	. 55	PUSH EBP
00401578	. 8BEC	MOV EBP,ESP
0040157A	. 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
00401590	. 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-10],ESP
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
004015A9	. 8BC8	MOV ECX,EAX
004015AF	. 81E1 FF000000	AND ECX,0FF
004015B5	. 890D D0524000	MOV DWORD PTR DS:[4052D0],ECX

  

Registers (FPU)	
EAX	0A280105
ECX	0A280105
EDX	00000001
EBX	7FFDC000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C910208 ntdll.7C910208
EIP	004015AF Malware_.004015AF
C 0	ES 0023 32bit 0(FFFFFFFF)
P 1	CS 001B 32bit 0(FFFFFFFF)
A 0	SS 0023 32bit 0(FFFFFFFF)
Z 1	DS 0023 32bit 0(FFFFFFFF)
S 0	FS 003B 32bit 7FFDF000(FFF)
T 0	GS 0000 NULL
D 0	
O 0	LastErr ERROR_INVALID_HANDLE (00000006)
EFL	00000246 (NO,NB,E,B,NS,PE,GE,LE)
ST0	empty -UNORM BDEC 01050104 005C0030
ST1	empty -UNORM BDEC 01050104 005C0030

Eseguendo uno “step-into” possiamo vedere che il valore di EXC cambia in **00000005**

00401575	. C9	LEAVE
00401576	. C3	RETN
00401577	. 55	PUSH EBP
00401578	. 8BEC	MOV EBP,ESP
0040157A	. 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
00401590	. 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-10],ESP
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
004015A9	. 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

  

Registers (FPU)	
EAX	0A280105
ECX	00000005
EDX	00000001
EBX	7FFDC000
ESP	0012FF94
EBP	0012FFC0
ESI	FFFFFFFF
EDI	7C910208 ntdll.7C910208
EIP	004015B5 Malware_.004015B5
C 0	ES 0023 32bit 0(FFFFFFFF)
P 1	CS 001B 32bit 0(FFFFFFFF)
A 0	SS 0023 32bit 0(FFFFFFFF)
Z 0	DS 0023 32bit 0(FFFFFFFF)
S 0	FS 003B 32bit 7FFDF000(FFF)
T 0	GS 0000 NULL
D 0	
O 0	LastErr ERROR_INVALID_HANDLE (00000006)
EFL	00000206 (NO,NB,NE,A,NS,PE,GE,G)
ST0	empty -UNORM BDEC 01050104 005C0030
ST1	empty -UNORM BDEC 01050104 005C0030

In questo caso viene eseguita l'istruzione AND sui bit ECX ed il valore 0FF (in esadecimale 255).