

S11L3

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.

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? (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)
3. 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).
4. BONUS: spiegare a grandi linee il funzionamento del malware.

SVOLGIMENTO:

1. Funzione CreateProcess

All'indirizzo 0040106E il malware effettua una chiamata di funzione alla funzione <CreateProcess>, come mostrato in figura sotto. Da notare i commenti nella colonna di destra immessi dallo stesso programma OllyDBG per una breve descrizione del malware.

0040103B	. 66 0C 74 5 D8 00 00	MOV WORD PTR SS:[EBP-28],0	
00401041	. 8B 55 18	MOV EDX,DWORD PTR SS:[EBP+18]	
00401044	. 89 55 E0	MOV DWORD PTR SS:[EBP-20],EDX	
00401047	. 8B 45 E0	MOV EAX,DWORD PTR SS:[EBP-20]	
0040104A	. 89 45 E8	MOV DWORD PTR SS:[EBP-18],EAX	
0040104D	. 8B 4D E8	MOV ECX,DWORD PTR SS:[EBP-18]	
00401050	. 89 4D E4	MOV DWORD PTR SS:[EBP-1C],ECX	
00401053	. 8D 55 F0	LEA EDX,DWORD PTR SS:[EBP-10]	
00401056	. 52	PUSH EDX	
00401057	. 8D 45 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 30 50 40 00	PUSH Malware_.00405030	
0040106C	. 6A 00	PUSH 0	
0040106E	. FF 15 04 40 40 00	CALL DWORD PTR DS:[<&KERNEL32.CreateProcessA>]	pProcessInfo
00401074	. 89 45 EC	MOV DWORD PTR SS:[EBP-14],EAX	pStartupInfo
00401077	. 6A FF	PUSH -1	CurrentDir = NULL
00401079	. 8B 4D F0	MOV ECX,DWORD PTR SS:[EBP-10]	pEnvironment = NULL
0040107C	. 51	PUSH ECX	CreationFlags = 0
0040107D	. FF 15 00 40 40 00	CALL DWORD PTR DS:[<&KERNEL32.WaitForSingleObject>]	InheritHandles = TRUE
00401083	. 33 C0	XOR EAX,EAX	pThreadSecurity = NULL
00401085	. 8B E5	MOV ESP,EBP	pProcessSecurity = NULL
			CommandLine = "cmd"
			ModuleFileName = NULL
			CreateProcessA
			Timeout = INFINITE
			hObject
			WaitForSingleObject

Come si può vedere dalla figura, il valore del parametro <CommandLine> è cmd.

2. Registro EDX

2.1: Qual è il valore del registro EDX?

Per questa richiesta è stato inizialmente inserito un breakpoint software (Toggle breakpoint) all'indirizzo 004015A3 e successivamente è stato eseguito il programma.

00401577	<M>	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:R1 00000000	MOV EAX,DWORD PTR FS:[0]	SE handler installation
0040158C		50	PUSH EAX	
0040158D		64:8925 00000000	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	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	
004015B8		C1E1 08	SHL ECX,8	
004015BE		03CA	ADD ECX,EDX	
004015C0		890D CC524000	MOV DWORD PTR DS:[4052CC],ECX	
004015C6		C1E8 10	SHR EAX,10	
004015C9		A3 C8524000	MOV DWORD PTR DS:[4052C8],EAX	
004015CE		6A 00	PUSH 0	
004015D0		E8 33090000	CALL Malware_.00401F08	
004015D5		59	POP ECX	
004015D6		85C0	TEST EAX,EAX	
004015D8		75 00	JNZ SHORT Malware_.004015F0	
EDX=00001DB1				
Malware_.<ModuleEntryPoint>+2C				

È stato quindi trovato il valore del registro EDX richiesto dall'esercizio, pari a **00001DB1**.

Registers (FPU)		<	<	<	<
EAX	1DB10106				
ECX	7EFDE000				
EDX	00001DB1				
EBX	7EFDE000				
ESP	0018FF5C				
EBP	0018FF88				
ESI	00000000				
EDI	00000000				
EIP	004015A3	Malware_.004015A3			
C 0	ES 002B 32bit 0(FFFFFFFF)				
P 1	CS 0023 32bit 0(FFFFFFFF)				

2.2 - 2.3: Eseguite a questo punto uno «step-into». Indicate qual è ora il valore del registro EDX motivando la risposta. Che istruzione è stata eseguita?

Successivamente è stato eseguito uno <step-into>, utilizzato per esaminare righe di codice e a fronte di una chiamata di funzione accedere alla sua implementazione. Ricontrollando il registro EDX ora si nota che il suo valore è cambiato, come mostrato in figura sotto.

Registers (FPU)		<	<	<	<
EAX	1DB10106				
ECX	7EFDE000				
EDX	00000000				
EBX	7EFDE000				
ESP	0018FF5C				
EBP	0018FF88				
ESI	00000000				
EDI	00000000				
EIP	004015A5	Malware_.004015A5			

Tale variazione è dovuta all'utilizzo dell'istruzione XOR logico. Tale operazione ritorna in output il valore 1 nel caso in cui i due valori di input siano diversi tra loro. Siccome l'operatore XOR è usato con gli input EDX ed EDX, l'output sarà sempre 0. Da cui il nuovo valore del registro EDX. Istruzione eseguita: XOR

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	

3. Registro ECX

Per le richieste del punto 3 è stato eseguito lo stesso procedimento del punto 2.

3.1: Qual è il valore del registro ECX?

Per questa richiesta è stato inizialmente inserito un breakpoint software (Toggle breakpoint) all'indirizzo 004015AF e successivamente è stato eseguito il programma.

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]	SE handler installation
0040158C	50	PUSH EAX	
00401590	64:8925 00000000	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-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	
004015B8	C1E1 08	SHL ECX,8	
004015BE	03CA	ADD ECX,EDX	
004015C0	890D CC524000	MOV DWORD PTR DS:[4052CC],ECX	
004015C6	C1E8 10	SHR EAX,10	
004015C9	A3 C8524000	MOV DWORD PTR DS:[4052C8],EAX	
004015CE	6A 00	PUSH 0	
004015D0	E8 33090000	CALL Malware_.00401F08	
004015D5	59	POP ECX	
004015D6	85C0	TEST EAX,EAX	
004015D8	75 00	JNZ SHORT Malware_.00401FE2	

ECX=1DB10106

Malware_.<ModuleEntryPoint>+38

È stato quindi trovato il valore del registro ECX richiesto dall'esercizio, pari a **1DB10106**.

Registers (FPU)		
EAX	1DB10106	
ECX	1DB10106	
EDX	00000001	
EBX	7EFDE000	
ESP	0018FF5C	
EBP	0018FF88	
ESI	00000000	
EDI	00000000	
EIP	004015AF	Malware_.004015AF

3.2 - 3.3 : Eseguite un step-into. Qual è ora il valore di ECX? Spiegate quale istruzione è stata eseguita.

Successivamente è stato eseguito uno <step-into>, utilizzato per esaminare righe di codice e a fronte di una chiamata di funzione accedere alla sua implementazione. Ricontrollando il registro ECX ora si nota che il suo valore è cambiato, come mostrato in figura sotto.

```

Registers (FPU)
EAX 1DB10106
ECX 00000006
EDX 00000001
EBX 7EFDE000
ESP 0018FF5C
EBP 0018FF88
ESI 00000000

```

Tale variazione è dovuta all'utilizzo dell'istruzione AND logico tra il valore del registro ECX ed il numero esadecimale FF.

```

00401580 | 8BC8 | MOV ECX, EAX
00401581 | 81E1 FF000000 | AND ECX, 0FF
00401585 | 8900 D0524000 | MOV DWORD PTR DS:[4052D00], ECX

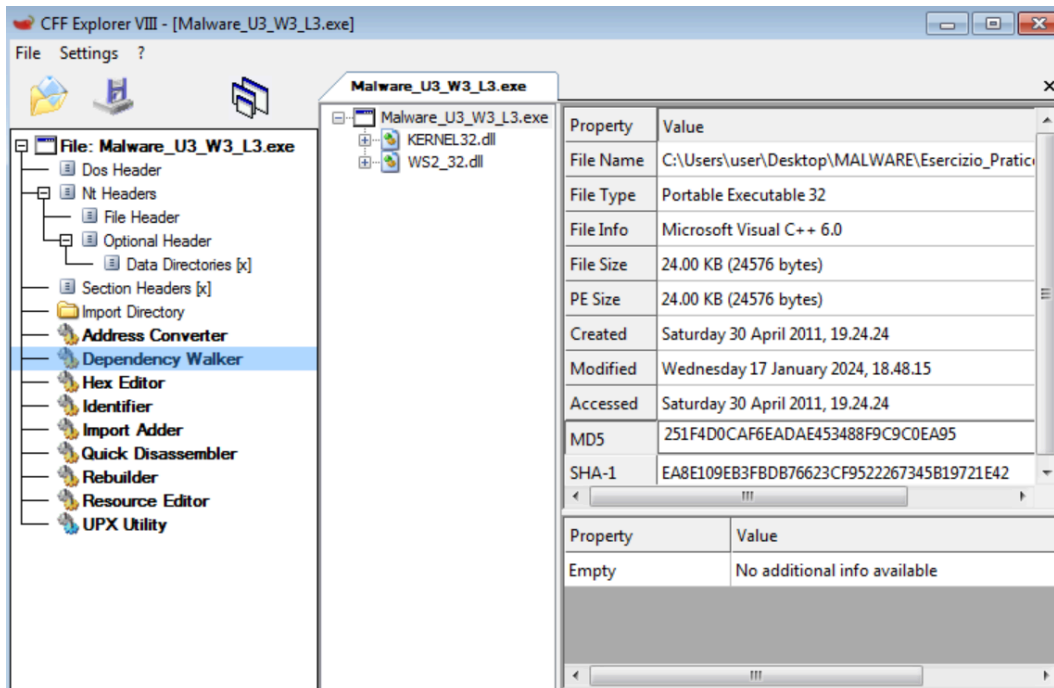
```

Il nuovo valore del contenuto del registro ECX è il risultato dell'operazione mostrata nella tabella seguente.

Operazione	Hex	Bin
AND	1DB1 0106	0001 1101 1011 0001 0000 0001 0000 0110
	FF	1111 1111
	0000 0006	0000 0000 0000 0000 0000 0000 0000 0110

4. Funzionamento del malware

Dopo questa analisi dinamica, ci siamo avvalsi di CFF Explorer per confermare le nostre ipotesi. Possiamo infatti intuire che il malware apra una CMD e vediamo con CFF Explorer importi le librerie Kernel32.dll e WS2_32.dll.



Per confermare il tipo di malware, usiamo Virus Total.

Il risultato ottenuto è stata l'identificazione del malware analizzato come un Trojan virus.

